

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-282432

(43)公開日 平成6年(1994)10月7日

(51)Int.Cl. ⁵	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 9/38	3 7 0 X	9193-5B		
	3 1 0 J	9193-5B		
15/16	3 9 0 Z	9190-5L		
15/66	K	8420-5L		

審査請求 未請求 請求項の数1 F D (全 25 頁)

(21)出願番号 特願平5-90614

(22)出願日 平成5年(1993)3月26日

(71)出願人 000000376

オリンパス光学工業株式会社

東京都渋谷区幡ヶ谷2丁目43番2号

(72)発明者 橋本 良昭

東京都渋谷区幡ヶ谷2丁目43番2号 オリ
ンパス光学工業株式会社内

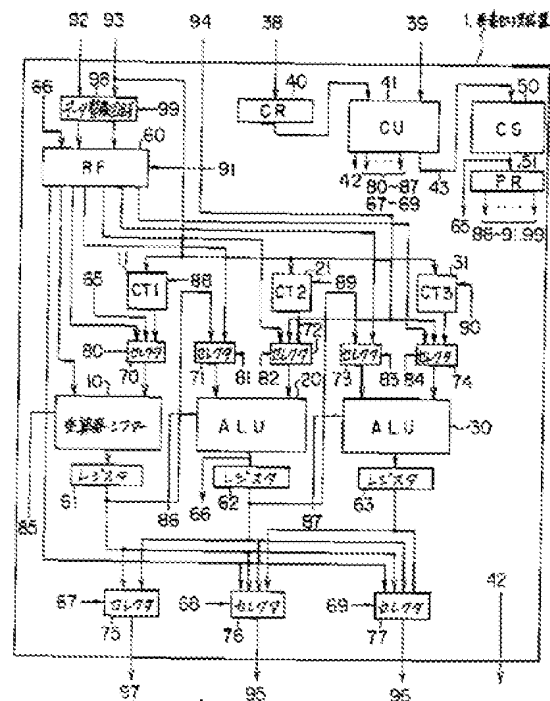
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 演算処理装置

(57)【要約】

【目的】用途に応じて適宜その構成を変更して装置規模を大きくせずに処理能力を向上させる演算処理装置を提供する。

【構成】複数のデータを同時に処理する演算処理装置において、異なる演算を同時に行なえる複数の演算手段10、20、30と、外部からのコマンドに応じて第1群の制御信号を生成する第1の制御手段41と、第1の制御手段41からの一部の制御信号に基づいて第2群の制御信号を生成する第2の制御手段50、51とを具備し、第1群及び第2群の制御信号に基づいて、複数の演算手段10、20、30の接続関係及びデータの流れを上記コマンドに応じた形態に適宜変更する。



【特許請求の範囲】

【請求項1】 複数のデータを同時に処理する演算処理装置において、異なる演算を同時に行なえる複数の演算手段と、外部からのコマンドに応じて第1群の制御信号を生成する第1の制御手段と、この第1の制御手段からの一部の制御信号に基づいて第2群の制御信号を生成する第2の制御手段と、を具備し、上記第1群及び第2群の制御信号に基づいて、上記複数の演算手段の接続関係及びデータの流れを上記コマンドに応じた形態に適宜変更することを特徴とする演算処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は演算処理装置に関し、特に、画像処理に適用される並列処理装置の形態と並列処理装置を利用した演算手法に関するものである。

【0002】

【従来の技術】 デジタル化された画像データに対して様々な処理を行なう画像処理には、フィルタリング、座標変換、直行変換等があるが、これらは多くは汎用の処理装置の上でプログラムにより処理される。

【0003】 1枚の画像を構成する画像データの数は膨大になるため、プログラム処理で対処できない場合は一部専用のハードウェアを用いるか、複数のプロセッサを並列に動作させることで処理能力の向上を狙っている。

【0004】 複数のプロセッサを並列に配置する場合は、汎用のプロセッサを2個並べたものから、プロセッサを画像を構成する画素データの数まで並べたものまである。

【0005】

【発明が解決しようとする課題】 しかしながら、汎用の処理装置でプログラムにより行うには時間がかかりすぎるという問題がある。また、画素データの一つ一つにプロセッサを割り当てた場合は規模が大きくなりすぎて一般に使用するものとしては不適當である。

【0006】 さらに、複数のプロセッサを並列に配置する場合はそれなりに処理能力の向上は期待できるが、期待したい性能を求めようとすると装置全体の規模が大きくなり、各プロセッサの細かな接続が難しくなるという問題がある。

【0007】 本発明の演算処理装置はこのような課題に着目してなされたものであり、その目的とするところは、用途に応じて演算手段の接続関係及びデータの流れを適宜変更することによって、装置規模を大きくせずに処理能力を向上させた演算処理装置を提供することにある。

【0008】

【課題を解決するための手段】 上記の目的を達成するために、本発明は、複数のデータを同時に処理する演算処

理装置において、異なる演算を同時に行なえる複数の演算手段と、外部からのコマンドに応じて第1群の制御信号を生成する第1の制御手段と、この第1の制御手段からの一部の制御信号に基づいて第2群の制御信号を生成する第2の制御手段とを具備し、上記第1群及び第2群の制御信号に基づいて、上記複数の演算手段の接続関係及びデータの流れを上記コマンドに応じた形態に適宜変更する。

【0009】

【作用】 すなわち、本発明の演算処理装置は、外部からのコマンドに応じて第1群の制御信号を生成すると共に、この第1群の制御信号の一部に基づいて第2群の制御信号を発生する。そして、第1群及び第2群の制御信号に基づいて、異なる演算を同時に行なえる複数の演算手段の接続関係及びデータの流れを上記コマンドに応じた形態に適宜変更する。

【0010】

【実施例】 以下図面を参照して本発明の実施例を詳細に説明する。

【0011】 図1は、第1実施例に係る要素処理装置1のブロック図である。

【0012】 本例においては1ワードのデータを32ビット長とする。また、以下の実施例で使用される“信号が有効となる”等の“有効”の意味は、対応する信号の出力が“1”となっている事を表すものとする。また、A(0:3)の様に記述した場合はA(0)、A(1)、A(2)の3本の信号またはレジスタを表すものとする。

【0013】 同図において、10は乗算器・シフター、11は乗算器・シフターの演算のリソースとなる定数等を格納するメモリ（以下CT1と呼ぶ）、20、30はALU、21はALU20のリソースとなる定数等を格納するメモリ（以下CT2と呼ぶ）、31はALU30のリソースとなる定数等を格納するメモリ（以下CT3と呼ぶ）、40は外部からのコマンドを格納するラッチ（以下CRと呼ぶ）。ここで、10は乗算器・シフター及びALU20、30は複数の演算手段を構成する。

【0014】 第1の制御手段としての制御部（以下CUと呼ぶ）41はCR40に保持されたコマンドの値と、要素処理装置1の外部からの信号39を入力として、必要な制御信号42、43、67～69、80～87を生成する。制御信号42は外部に対し、動作タイミングやエラーを知らせる為のものである。制御信号43はメモリ（以下CSと呼ぶ）50のアドレスとなり、CS50の出力信号はラッチ（以下PRと呼ぶ）51に保持され、PR51の出力は制御信号として使用される。ここで、CS50とPR51とは第2の制御手段を構成する。

【0015】 また、複数のレジスタ（以下RFと呼ぶ）60は要素処理装置1外部からのデータを保持する。本

3

例においてRF60は8ワードのレジスタから構成されるものとし、それぞれをRF60(0)、RF60(1)、…、もしくはRF(0)、RF(1)、…、のように表すこととする。RF60は入力、出力合わせて7ポートを同時に動作できるものとする。

【0016】レジスタ61は乗算器・シフター10の出力を格納するもので2ワード長である。レジスタ62はALU20の出力を保持し、レジスタ63はALU30の出力を保持する。図中、70、71、72、73、74、75、76、77はデータを選択するためのセクタであり、セクタ70~74はCU41の出力信号80~84で、セクタ75~77は出力信号67~69により制御される。

【0017】CU41の出力信号85は乗算器・シフター10の制御を、出力信号86はALU20の制御を、出力信号87はALU30の制御を行う。CT1(11)、CT2(21)、CT3(31)の制御を行う制御信号88~90と、RF60の制御を行う制御信号91はPR51より出力される。

【0018】要素処理装置1にはデータ入力図中92、93、94の3種類、データの出力図中95、96、97、の3種類がある。図中98はデータの相合わせ等を行うデータ整備回路である。データ整備回路98はPR51の出力信号99により制御される。

【0019】次に要素処理装置1の各部の動作を定義する。

【0020】各部は大きく2つの種類に分ける事ができる。1つはCR40の中に格納されたコマンドで一意的に決まるものと、コマンド実行中に各部がサイクル毎に変化するものである。まず一意的に決まるものはセクタ70~77で選択するデータ及び、乗算器・シフター10、ALU20、ALU30の演算動作、データ整備回路98のデータの扱い方である。サイクル毎に変化するものはRF60、CT1(11)、CT2(21)、CT3(31)の各アドレスである。それぞれに対しての制御コードを図24~図29のように設定する。すなわち、図24は乗算器・シフター10の制御コードMOP(信号85)であり、図25はALU20、30の制御コードAOP(信号86、87)であり、図26はセクタ70~77の制御コードS2OP、S3OP(信号80~84、信号67~69)であり、図27はCT1(11)~CT3(31)の制御コードTOP(信号88~90)であり、図28はRF60の制御コードROP(信号91)であり、図29はデータ整備回路98の制御コードFOP(信号99)である。図32にRF60の入出力信号とポートの関係を示す。入出力信号の接続は図1の接続位置と同じである。RF60への書き込みができるのはポートa、d、fの3ポートである。図24~図29の各部の制御コードから分かるように、CU41から出力する制御信号の合計は33ビット

4

となり、PR51から出力される制御信号の合計はRF60の書き込みが最大3ポートであることより39ビットとなる。

【0021】本要素処理装置1はスレーブプロセッサとして動作する事を前提としている。このために、図3に示すように、要素処理装置1の外部にホストとなる制御プロセッサ55を設ける必要がある。図中55が制御プロセッサである。要素処理装置1が処理すべきデータも制御プロセッサ55から与えられ、処理結果も制御プロセッサ55に返される。

【0022】要素処理装置1の動作を説明する。まず制御プロセッサ55により、初期化が行われる。制御プロセッサはCR40に初期化コマンドを格納するとともにデータ信号92、93、94に“0”を出力する。要素処理装置1はこのコマンドによりCU41内部のレジスタ、フリップフロップ等を制御に必要な値をセットし、初期化コマンドに対応したCU41内の順序回路が起動し、CS50から初期化に必要な信号を出力するためのアドレス43を生成し出力する。

【0023】CS50はアドレス43でアドレスされる内容を順次出力し、CT1(11)、CT2(21)、CT3(31)及びRF60にデータ92、93、94の値を格納することでメモリ、レジスタ類の初期化を行う。制御プロセッサ55は初期化に必要な時間をカウントすることで初期化の終了を認識する。

【0024】今、制御プロセッサ55は2種類の計測データDATA1(0:1000)とDATA2(0:1000)から、同一の番号を持ったデータどおしの積M(0:1000)と、

【0025】

【数1】

$$S = \sum_{n=0}^{999} M$$

を求める。ここで、nを求める処理は要素処理装置1が行うものとする。RF60はRF(1)をDATA1の受け渡しに、RF(2)をDATA2の受け渡しに、RF(3)をSの中間結果の保持に使用する。乗算器・シフター10でRF(1)とRF(2)の値の積を計算し、結果をセクタ76を経て、信号95として制御プロセッサ55に出力する。ALU20で乗算器・シフター10の計算結果とRF(3)の値を加算し、加算値をRF(3)に格納する。

【0026】加算結果はセクタ77を経て信号96として出力される。これより、制御プロセッサ55はSの中間結果も知る事ができる。各セクタの選択するリソースはセクタ70がA、セクタ71がA、セクタ72がA、セクタ76がB、セクタ77がCである。これ以外のセクタはどのリソースを選択していても処理には関係ない。またALU30も使用しない。処

5

理に関係ないもの、使用しないものに対しての制御コードは最も値の小さいコードが出力されるものとする。

【0027】上記演算を実行するためのコマンドをデータ数をmとして“MAA(m)”という記号で表す事にする。制御プロセッサ55はコマンド“MAA(1000)”に対応するコマンドコードをCR40にセットする。CR40の出力はCU41に入力され、CU41はコマンド“MAA(1000)”に対応した制御信号67~69、80~87とCS50のアドレス43を生成する。

【0028】このときの制御信号の値を図4に示す。これにより各セクタ、乗算器・シフター10、ALU20の動作が決定される。アドレス43によりCS50の内容が出力され、PR51に保持される。この時のPR51に保持された値を図5(A)に示す。この状態で制御プロセッサ55はデータ入力信号線92にDATA1(0)を、信号線93にDATA2(0)を出力する。1サイクル実行後、DATA1(0)はRF(1)に、DATA2(0)はRF(2)に格納される。

【0029】乗算器・シフター10は乗算を、ALU20は加算を実行するが、RF(3)への書き込みは許可されていないため、RF(3)は初期設定で“0”となったままである。データ出力信号95、96には乗算結果、加算結果が出力されるが制御プロセッサ55は最初の結果は無視をする。1サイクル実行した事でアドレス43は次の信号の入ったアドレスを指す。

【0030】制御プロセッサ55は次のデータであるDATA1(1)、DATA2(1)を信号線92、93に出力する。要素処理装置1内の制御信号によってアドレス43とCS50の出力、PR51の出力が変化する事になるが、このコマンドの場合、2サイクル目も同じ制御コードが出力される。この状態で2サイクル目が実行される。RF(1)にはDATA1(1)が、RF(2)にはDATA2(1)が格納され、レジスタ61にはDATA(0)とDATA2(0)の積が格納される。

【0031】この値は信号線95で制御プロセッサ55に伝えられ、まず制御プロセッサ55は2サイクル目で最初のデータの積を知る事ができる。1サイクル実行した事でアドレス43が変化する。これによりCS50の出力が図5(B)のように変化する。変化したのはポートaに対しての制御コードROP(0)だけである。これにより次のサイクルからRF(3)へのデータの格納が行われる。制御プロセッサ55は次のデータであるDATA1(2)を信号線92に、DATA2(2)を信号線93に出力する。

【0032】この状態で3サイクル目が実行される。レジスタ61にはDATA1(1)とDATA2(1)の積が格納される。この値は信号線95により制御プロセッサ55に出力される。レジスタ62にはレジスタ61

6

の値とRF(3)の値の和が格納される。RF(3)は“0”のままであるからレジスタ62はDATA1(0)とDATA2(0)の積がそのまま格納されることになる。またこの値は信号66によりRF60のポートaの入力となり、RF(3)に格納される。レジスタ62の値は信号線96により制御プロセッサ55に出力され、これにより制御プロセッサ55はSの中間結果を知る事になる。

【0033】以降はCU41によりアドレス43の値は固定され、PR51の保持する値も変化しない。CU41はコマンドMAA(1000)で指示されたデータ分だけサイクルを繰り返す。制御プロセッサ55は新たなデータを出力する。これにより、制御プロセッサ55は、スタートしてから1001サイクル目に最終データであるDATA1(999)とDATA2(999)の積を、1002サイクル目でSの値を知る事になる。

【0034】処理の途中、制御プロセッサ55で何らかの都合により、データ出力のタイミングが合わなくなった等の不都合が生じた場合は信号39により通知し、CU41は要素処理装置1内の必要な所にWAIT信号(図示なし)を出力し、処理を待たせ、信号39が解除されたところで処理を再開する。オーバーフロー等のエラーが発生したときにエラー信号(図示なし)はCU41に集められ、1つの信号42にまとめられ、制御プロセッサ55に通知される。

【0035】図24で示しているMOP(2:2)、図25で示しているAOP(3:2)は乗算器・シフター10、ALU20、30の演算時のデータ幅を規定するものである。それぞれ“00”の時は32ビット、“01”の時は16ビット、“10”の時は8ビットとなる。16ビット指定の時は、乗算器・シフター10、ALU20、30が上位DATA16ビットと下位16ビットで半分に分けて動作し、キャリー及びボローの伝達が行われず、シフトの時も別個の動作となる。データラインも分割して使用する。8ビット指定の時は同様にして4分割し、使用する。

【0036】ひとつのコマンドの処理の中で、セクタ70~77の切り替えや、乗算器・シフター10、ALU20、30の演算の種類を変更したいような場合はCS50の出力ビット幅を増やし、制御信号80~87、67~69の信号値もCS50に格納するように変更することで対応が可能である。

【0037】第1実施例によれば、汎用のプロセッサと違って、内部の構成要素の接続形態を処理すべきコマンドに対応して変更するので、処理内容に適した接続形態とすることができ、効率のよいALUとすることが可能となる。また、同じ要素処理装置の並列性の制御がしやすくなる。

【0038】図6は、本発明の第2実施例に係る基本処理装置100のブロック図である。

【0039】図中110、120は前記した要素処理装置1と同様の構成を有する。150は2つの要素処理装置の同期、エラー制御等を行う為の同期制御回路（以下SUと呼ぶ）である。111、121は2つの要素処理装置110、120の出力からそれぞれ1つを選択するためのセクタである。140、141は選択されたデータの出力である。153はセクタの制御信号であり、SU150で生成される。信号130は図1におけるデータ入力信号92に、131はデータ入力信号93に、132は信号38に対応する。信号133は外部からの同期及び制御の為の信号の入力である。

【0040】本例では信号133は3ビットとし、その意味を図30に示す。SU150からの出力である信号151と信号152は図1の信号39に対応する。要素処理装置1の動作説明では信号39の意味の設定は行わなかったが、本例では図31に示す。SU150の入力となる信号112、122は図1の信号42に対応する。

【0041】図6の113、123は図1のデータ出力信号の95に、114、124はデータ出力信号96に、115、125はデータ出力信号97に対応する。信号142は基本処理装置100内で発生したエラー等の情報を外部に伝えるためである。ここで、基本処理装置100もスレーブプロセッサとして動作するものとして説明する。

【0042】図7は基本処理装置システムのブロック図であり、要素処理装置1の場合と同様に制御プロセッサ55に直接接続することとする。制御プロセッサ55は基本処理装置100に次の処理を実行させる。すなわち、2種類の観測データDATA1（0:1000）、DATA2（0:1000）をそれぞれ別個の定数を乗じ、補正の為の数値の加算を行った後、2つの値の和と差を求める。

【0043】要素処理装置110、120はデータを受け取った後、乗算と加算を2回または乗算と加算の後減算を行う。乗算と加算を2回行うことを指示するコマンドは“MAW(n)”、乗算と加算の後減算を行うことを指示するコマンドは“MAS(n)”で表すものとする。nはデータ数である。

【0044】制御信号133で送られる信号は1サイクル後の動作に対しての意味を伝える。まず制御プロセッサ55は信号133にコマンド設定の為の信号値“100Z”を出力する。次に制御プロセッサ55は信号133を“111Z”にするとともに、データ信号130、131に“0”を出力し、初期化コマンドを信号線132に出力する。これにより要素処理装置110、120は初期化され、同期制御部SUも初期化される。

【0045】制御プロセッサ55は初期化に要する時間をカウント後、信号133に“101Z”を出力する。次のサイクルに信号132に“MAW(1000)”に

相当するコードを出力し、信号133に“110Z”を出力する。次に信号132に“MAS(1000)”に相当するコードを出力し、信号133に“111Z”を出力する。この3サイクルで要素処理装置110のCR40にコマンド“MAW(1000)”が、要素処理装置120のCR40にコマンド“MAS(1000)”が格納される。

【0046】制御プロセッサ55からのデータは各要素処理装置のRF(1)に格納するものとし、乗数、補正の為の加算定数はそれぞれ各要素処理装置のCT1(11)、CT2(21)に格納されているものとする。乗数はDATA1に関してが“a”、DATA2に関してが“b”、補正の為の加算定数はDATA1に関してが“c”、DATA2に関してが“d”とする。要素処理装置110のCT1(11)のアドレス“0001Z”に“a”を、CT2(21)のアドレス“0001Z”に“c”を格納する。要素処理装置120のCT1(11)のアドレス“0001”に“b”を、CT2(21)のアドレス“0001Z”に“d”を格納する。

【0047】要素処理装置110では信号130で送られてくるDATA1の補正を、要素処理装置120では信号131で送られてくるDATA2の補正を行うものとする。これらCT1(11)、CT2(21)のアドレスやデータを入力する信号の組み合わせはコマンドの種類により予め決まるものである。

【0048】要素処理装置110内のセクタは次の様に設定される。セクタ70はリソースCを、セクタ71はリソースAを、セクタ72はリソースCを、セクタ73はリソースAを、セクタ74はリソースBを、セクタ75はリソースBを、セクタ76はリソースDを選択する。信号96へは本処理では必要なデータを出力しないため、リソースAが選択される。

【0049】要素処理装置110の処理は、乗算器・シフター10でRF(1)と“a”の乗算を、ALU20はレジスタ60の値と“c”の加算を、ALU30ではレジスタ62と要素処理装置120でDATA2の補正されたデータを信号125で受け取り加算を行う。

【0050】なお、要素処理装置120内のセクタの設定は、要素処理装置110と同じである。

【0051】要素処理装置120の処理は、乗算器・シフター10でRF(1)と“b”の乗算を、ALU20ではレジスタ61の値と“d”の加算を、ALU30ではレジスタ62と要素処理装置110からのDATA1の補正データを信号115で受け取り減算を行う。

【0052】以下に第2実施例の動作を説明する。要素処理装置110は次のように動作する。

【0053】1サイクル目で制御プロセッサ55から信号130として受け取ったDATA1をRF(1)に格納する。

【0054】2サイクル目で乗算器・シフター10で乗

算を行い、結果がレジスタ61に格納される。RF (1)には制御プロセッサ55からの次のデータが送られてくる。

【0055】3サイクル目で乗算器・シフター10で2つ目のデータの乗算を、ALU20で1つ目のデータの加算を行う。乗算結果はレジスタ61に、加算結果はレジスタ62に格納される。RF (1)には制御プロセッサから次のデータが送られてくる。レジスタ62の値はセクタ75を経て信号115として要素処理装置120に送られる。

【0056】4サイクル目で乗算器・シフター10で3つ目のデータの乗算を、ALU20で2つ目のデータの加算を、ALU30でレジスタ62に格納された補正された1つ目のデータと要素処理装置120でDATA2の補正された値との加算を行う。3つ目のデータの乗算結果はレジスタ61に、2つ目のデータの加算結果はレジスタ62に、1つ目のデータの加算結果はレジスタ63に格納される。

【0057】RF (1)には制御プロセッサ55から次のデータが送られ、2つ目のデータの補正された値は要素処理装置120に送られる。1つ目のデータは最終的な結果である、補正された値どうしの和となりセクタ76を経て制御プロセッサ55に出力される。

【0058】要素処理装置120は要素処理装置110と同じタイミングで動作するが、要素処理要素110との違いは信号131からのDATA2をRF (1)に格納する事とALU30で減算を行うことである。従って要素処理装置110で2つの補正値の加算結果がでると同時に要素処理装置120では減算結果が出る事になる。

【0059】第2の実施例によれば、同一のデータから2種類の結果を得る場合の接続性が良い。また、要素処理装置間の接続も変えられる構成とすることで一層の柔軟性を持たせることができる。

【0060】図8は本発明の第3実施例に係る演算処理装置200の構成図である。同図において、基本処理装置240~243 (P0, P12, P3, P15)、250~253 (P1, P2, P13, P14)、260~263 (P5, P9, P6, P10)は前記した基本処理装置100と同様の構成を有する。ここで、基本処理装置Pxxの内部の要素処理装置は、図6中左側にあるものをPxx-0、右側にあるものをPxx-1と呼ぶことにする。なお、同図のごとく接続された形態は一般に4個×3段のオメガ網と呼ばれる。図8においての各要素処理装置の入出力信号は、図6の信号133と信号142以外は接続位置がそのまま対応する。信号133と対応するのは信号282、信号142と対応するのは信号281である。信号281と282は図上では1つの信号として記述しているが、基本処理装置ごとに独立した信号線が接続されている。220~227、23

0~237はセクタ、270~277はローカルメモリ (以下LM0~LM7と呼ぶ) である。LM0~LM7にあるLMCは、LM0~LM7を制御するコントローラであり、最終段の基本処理装置はデータに付加するタグでLMCに動作を指示する。セクタ220~227は2つのリソースから1つを選択するレジスタであり、セクタ230~237は図では省略した記述となっているが、8つのリソースから1つを選択するセクタである。

10 【0061】210は外部とのデータの受け渡しの為の高速バスインターフェイスである。この高速バスインターフェイスは例えば、132Mbps程度の転送速度を有する。外部からくる信号としては信号290、291がある。信号290はコマンド信号であり、信号291は外部との同期等をとるための信号である。また外部に対する同期、エラー等の為の出力信号が信号292である。280は各基本処理装置、及びセクタ220~227、230~237の切り替え、高速バスインターフェイス210の制御、LM0~LM7 (270~277)の制御を行う制御部 (以下MCUと呼ぶ) である。

20 【0062】演算処理装置200もスレーブプロセッサとして動作するものとして説明する。図9に制御プロセッサ55と接続した図を示す。データは高速バス211を介して入出力され、処理に必要なコマンドは信号290により、必要な制御は信号291で与えられ、エラー等は信号292で制御プロセッサ55に知らされる。

30 【0063】基本処理装置240~243、250~253、260~263の動作は既に説明した通りである。これら4個×3列のオメガ網に接続した基本処理装置は、基本処理装置100の動作説明時に図7の制御プロセッサ55から受けたのと同じ制御をMCU280から受ける。

40 【0064】コマンド設定時、制御プロセッサ55はコマンドを信号290に出力する2サイクル前に、コマンドを設定する基本処理装置の番号とともに信号291によりMCU280に指示する。MCU280は対応する基本処理装置に信号282を介して既に基本処理装置の動作の説明をした信号値を出力し、コマンドの設定を行う。すべての基本処理装置についてのコマンドの設定が終了すると、MCU280に対しての動作モードが信号290で指示される。MCU280はこれにより、これから行う処理に対応したセクタ220~227、230~237やLM0~LM7 (270~277)、高速バスインターフェイス210の制御を行う。

50 【0065】制御プロセッサ55は高速バス211にデータを出力し、演算処理装置200はこのデータを受け取り、各基本処理装置は設定されたコマンドの処理を行い、順次後段の基本処理装置に伝達する。最終段の基本処理装置260~263の演算結果はMCU280の制御によりLM0~LM7 (270~277)に格納され

るか、高速バスインターフェイス210、高速バス211を経て制御プロセッサ55に返される。LM0~LM7(270~277)の内容は再び1段目の基本処理装置240~243の入力となる事もある。

【0066】以上のようにして本演算処理装置200は指示された演算を行う。

【0067】第3の実施例によれば、4個×3列のオメガ網に接続した基本処理装置を1つの単位とすることで、システムの構成をしやすくなるとともに、要素処理装置から見ると完全シャフル型となることでバタフライ演算を高速に行うことができる。また、データの流れが1方向であり効率が低下する点もあるが、その分はデータの移動の制御が簡単であること、適したアルゴリズムをとることで補うことができる。

【0068】次に本発明の第4実施例を説明する。この実施例は前記した演算処理装置200を画像処理装置に適用したものである。

【0069】図2は第4実施例の画像処理装置300のブロック図である。図中301は必要なプログラム、データを格納するメモリモジュールである。303は外部からのアナログ信号を装置内で処理できるようにデジタル信号に変換する回路と装置内部で処理を行ったデジタル信号をアナログ信号に変換する回路を持ったA/D・D/A変換回路である。304は本装置全体の制御を行う制御プロセッサ、305は前記した演算処理装置200に対応する演算処理装置である。演算処理装置305は増設が可能であり、同図の点線で囲まれた部分が示すように、シリアルあるいはパラレルに接続することができる。処理性能は増設にともないほぼ線形に向上する。

【0070】I/Oプロセッサ310はファイル装置311、プリンタ312、端末装置313と接続され所定の処理を行う。オペレーターは端末装置313を使用して画像処理装置300の操作を行う。302はメモリモジュール301、A/D・D/A変換回路303、制御プロセッサ304、演算処理装置305を接続する高速バスである。高速バス302は32ビットの幅のデータラインを8CHもち、メモリモジュール301はそれぞれ対応して存在する。

【0071】8CHのデータラインは独立に動作し、演算処理装置305の高速バスインターフェイス210(図8参照)で転送されて基本処理装置に割り振られる。高速バスインターフェイス210は高速バスの動作サイクルと演算処理内部の動作サイクルの整合を取るための緩衝装置としての役割も持ち内部にはキャッシュメモリに相当するデータ保持部を設けても良い。

【0072】309は制御プロセッサ304とI/Oプロセッサ310を接続するI/Oバスである。307は画像処理装置300外のアナログ信号の生成元としてのCCDカメラ、308は画像処理装置300で処理した結果を見るためのモニターである。

【0073】画像処理装置300は、例えば、CCDカメラ307で撮影した画像データの連続した4枚分の同位置の画素データの平均値を求めるべくノイズ除去と強調を行う。画像データは図10のように、画像のサンプリング時間を t とすると、ある時刻 T でサンプリングした画像データの左半分の画素データはメモリモジュール301のCH0に、右半分の画像データはCH2に、時刻 $T+t$ でサンプリングした画像データのうち左半分の画像データはCH1に、右半分の画像データはCH3に格納される。また、時刻 $T+2t$ でサンプリングした画像データの左半分はCH4に、右半分の画像データはCH6に、時刻 $T+3t$ でサンプリングした画像データの左半分の画像データはCH5に、右半分の画像データはCH7に格納される。

【0074】時刻 $T+4t$ のサンプリングデータは時刻 T におけるメモリモジュール301の内容に上書きされる。サンプリングデータのメモリモジュール301への格納はA/D・D/A変換回路303が行っている。

【0075】上記平均値を求める動作の指示はオペレーターが端末装置313より行い、I/Oプロセッサ310を経由して制御プロセッサ304に伝えられる。制御プロセッサ304はこれを受けてA/D・D/A変換回路303、演算処理装置305にコマンドの設定等必要な処理を行う。

【0076】演算処理装置305内の基本処理装置P0(図8)はメモリモジュールCH0、CH1から読みだした同位置の同画素データの加算を行う。同様に、P1はCH2、CH3のデータを、P2がCH4、CH5のデータを、P3がCH6、CH7のデータの加算を行う。P4ではP0とP3の出力結果の加算を、P5ではP1とP2の出力結果の加算を行う。P6はP4の出力結果を右に2ビットシフトし出力し、P7はP5の出力結果を右に2ビットシフトし出力する。

【0077】処理結果はメモリモジュールCH0とCH1に入力データと重ならないようにして格納され、A/D・D/A変換回路303に読み出されモニター308に映し出される。演算処理装置305で必要なメモリモジュールへのアクセスは演算処理装置305内のMCU280で行われる。処理は演算処理装置305の1サイクルで2画素分が行われる。

【0078】このように、第4の実施例によれば、高速の画像処理装置が構成できる。なお、要素処理装置、基本処理装置、演算処理装置は上述した方法でしか使用できないものではなく、それ自体で別な使い方ができる。

【0079】上記した実施例では必ずしもすべての基本処理装置、要素処理装置が動作する訳ではない。従ってこれら使用されない部分を使用することによって信頼性を高める実施例を以下に説明する。

【0080】図11は本発明の第5実施例に係る基本処理装置100のブロック図である。117、127は信

号115と125の比較を行う比較器である。比較結果は信号116、126で制御部（以下SUと呼ぶ）150に出力する。不一致を検出した場合はSU150はエラーを信号142により外部に出力する。検出器117、127の結果を有効とするかどうかはSU150内にフラグを設け、このフラグのオン/オフにより判断するものとし、このフラグは信号133により制御できるようにする。

【0081】また図示はしていないが、図1においてはセクタ75は2つのリソースから1つを選択するものであるが、図11のものはレジスタ63の出力もリソースに加え、3つのリソースから1つを選択するものに変更してある。上記機能を盛り込んだのが要素処理装置118、128である。

【0082】今、要素処理装置118が必要な処理を行っているものとする。この時に要素処理装置118のセクタ75を比較を行いたい信号に切り替えておき、要素処理装置128に演算も含めて同様の動作となるようコマンドをセットする。この状態で処理を行えば比較器117、127で常時演算結果のチェックが可能となり、信頼性を高める事ができる。

【0083】比較する対象も各要素処理装置の最終的なものだけでなく、セクタ75を切り替えるだけで、乗算器・シフター10（図1）の演算結果、ALU20の演算結果、ALU30の演算結果と選択することが可能である。

【0084】以下に、図12を参照して本発明の第6実施例を説明する。これまで説明してきた実施例において、各基本処理装置に対してのコマンドの設定は制御プロセッサにより設定する基本処理装置を選択し、対応するコマンドを与えることで行ってきた。処理すべきものが装置して決定されており、コマンドの入れ替えの頻度が少ない場合はこれでもよいが、コマンドの設定を頻繁に行うような場合、演算処理装置1台でも最大で26サイクルも要するため総合的な処理速度に対して影響が出てくると考えられる。本演算処理装置はMIMDであるが、これをSIMDとして扱えるようにする事によりこれを解決する。

【0085】図12は縦に接続した2つの要素処理装置を示したものである。図中の数字についている“'”は基本的に同じものであるが異なった構成を有するので識別のためにつけたものである。図中48及び48'は新たに設定した要素処理装置番号を格納するレジスタである。レジスタ48への値の設定は装置の初期化の時に制御プロセッサにより行う。CU41、41'及びCS50、50'は演算処理装置200の行うすべての処理に対してCR40の内容とレジスタ48（48'）とから制御信号を生成するように作り直されたものである。図12中の198、198'は図示を簡単にするためのものであり、図1のCR40、CU41、CS50、PR

51以外の部分をまとめたものである。

【0086】今、上段の要素処理装置を図8における基本処理装置P1を考慮してP1-0、図8のP5を考慮して下段の要素処理装置をP5-0とする。

【0087】演算処理装置の実施例で説明したのと同じ動作をしているとすると、以前のままの要素処理装置の場合、P1-0は信号92と93で伝えられたデータを加算し、その結果がセクタ111で選択されP5-0の入力となり、P5-0ではこれを2ビットシフトして信号95'として出力している。このときCR40に設定されているのは加算をするのとこれに必要なセクタを指示するコマンドであり、CR40'に設定されているのは2ビット右シフトをするのとこれに必要なセクタの選択を指示するコマンドである。

【0088】4枚の画像データから対応する画素値の平均を算出するコマンドを“AV4”とし、これに対するコマンドコードをCR40、40'に同時に設定すると、各要素処理装置はこのコマンドコードと48、48'に格納されて要素処理装置番号より判断し、P1-0では信号92と93で与えられたデータを加算し、出力するための制御信号がCU41及びCS50から出力される。P5-0では信号92で伝えられる値を2ビット右シフトし、信号95'として出力するための制御信号が出力される。

【0089】これにより、すべての要素処理装置に設定するコマンドは同じものでも各要素処理装置に異なった動作をさせる事ができる。

【0090】第6の実施例によれば、演算処理装置内の基本処理装置に対して同一のコマンドが与えられても、それぞれの基本処理装置は要求される異なる処理を行うため、コマンド制御が簡単になる。

【0091】以下に図1及び図8を参照して本発明の第7実施例について説明する。

【0092】図8の4個×3段に接続された基本処理装置を上から1段目、2段目、3段目と呼ぶことにする。そして、32ビット分の4個の8ビットデータを演算処理装置200に入力し、上位2個分のデータを1段目の基本処理装置で処理し合成して、未処理のデータと合わせて1つの32ビットデータとして2段目に出力する。次に、2段目の基本処理装置は残った下位データの内の1つについての処理を行い、前段から受け取った処理結果と合成し、未処理の1個の8ビットデータと合成して1つの32ビットデータとして3段目の基本処理装置に出力し、3段目の基本処理装置は残った未処理のデータを処理し、前段から受け取った処理結果と合成し、最終の処理結果を作成する。

【0093】すなわち、本実施例においては、図1の信号85、86、87に新たにデータマスク信号DMS（0：4）を追加する。DMS（0：4）は上位ビットから入力データの上位バイトに対応する。DMS（0：

4) のビットが有効になっている所の入力データのバイトが演算で使用され、処理結果はバイト単位で左につめて出力する。

【0094】演算の為に使用する8ビットデータを $a_0 \sim a_{15}$ 、 $b_0 \sim b_{15}$ で表し、この2種類のデータ間の演算を行う事とする。演算結果を $z_0 \sim z_{15}$ で表す。

【0095】初期化、及び各部に対してのコマンドの設定は終了しているものとし、各基本処理装置の入力となるデータは図で左側にしめされたものをx入力、右側に示されたものをy入力と呼ぶことにする。

【0096】2種類のデータは、

P0 のx入力 a_0, a_1, a_2, a_3

y入力 b_0, b_1, b_{10}, b_{15}

P12のx入力 a_4, a_5, a_6, a_7

y入力 b_4, b_5, b_{14}, b_{11}

P3 のx入力 a_8, a_9, a_{10}, a_{11}

y入力 b_8, b_9, b_2, b_7

P15のx入力 $a_{12}, a_{13}, a_{14}, a_{15}$

y入力 b_{12}, b_{13}, b_6, b_3

となるように高速バスインターフェイス210に要求する。

【0097】本例を説明するに当たって制御コードAOP(0:3) = "111Z"のFNCにおける特定処理を定義する。AOP(3:3) = "001Z"のとき、ALU20、30は、"-"をデータの合成を示す接続子とすると、1段目の基本処理装置内の要素処理装置0、1(以下、図6の110、120に対応するものとする)は、

左入力データ(0:8) - 右入力データ(8:24)

2、3段目の基本処理装置内の要素処理装置0、1は、

右入力データ(0:8) - 左入力データ(0:16)

- 右入力データ(24:8)

のようにデータの合成を行う。

【0098】各要素処理装置内の制御コードはMOP(2:2) = "10Z"、ALU20に対してAOP(2:2) = "10Z"、ALU30に対しては、2段目、3段目の基本処理装置内の要素処理装置1にはNOPを、これ以外の要素処理装置にはAOP(0:6) = "111001Z"を出力する。MOP(0:2)、MOP(4:2)とALU20に対してのAOP(0:3)、AOP(5)は処理に必要な制御信号を出力する。マスク信号DMS(0:4)は1段目の基本処理装置内の要素処理装置0は信号85、86が"1000Z"、信号87が"1111Z"、要素処理装置1は信号85、86が"0100Z"、信号87が"1111Z"、2段目の基本処理装置内の要素処理装置0は信号85、86に"0010Z"、信号87に"1111Z"、要素処理装置1は信号85、86、87ともに"1111Z"、3段目の要素処理装置0は信号85、

86が"0001Z"、信号87は"1111Z"、要素処理装置1は信号85、86、87とに"1111Z"である。

【0099】入力データは各要素処理装置内のRF(6)とRF(7)に格納され、処理が行われる。処理は要素処理装置P0-0で z_0 、P0-1で z_1 、P12-0で z_4 、P12-1で z_5 、P3-0で z_8 、P3-1で z_9 、P15-0で z_{12} 、P15-1で z_{13} 、P1-0で z_2 、P2-0で z_{10} 、P13-0で z_6 、P14-0で z_{14} 、P5-0で z_3 、P9-0で z_7 、P6-0で z_{11} 、P10-0で z_{15} の演算を行い、最終的にP5の出力信号95に z_3 、 z_{13} が、出力信号96に z_0 、 z_2 が、P9の出力信号95に z_7 、 z_9 が、出力信号96に z_4 、 z_6 が、P6の出力信号95に z_8 、 z_{11} が、出力信号96に z_5 、 z_{14} が、P10の出力信号95に z_{12} 、 z_{15} が、出力信号96に z_1 、 z_{10} が出力され、セクタ230~237で並べ変えられる。各要素処理装置における入出力データを図13に示す。

【0100】本実施例ではデータのマスクや合成にALUを使用した、セクタの制御コード数を増やし、セクタで行う方法もある。

【0101】第7実施例によれば、ハードウェアの無駄がなくなり、処理能力も最大で4倍となる。

【0102】以下に、図14を参照して、積和演算を行う本発明の第8実施例について説明する。

【0103】第1に、本実施例の構成は、演算処理装置を使用して 3×3 の積和演算を行うために、図8の演算処理装置200と同様の構成を有する演算処理装置400と、データを格納するメモリD402と、演算処理装置400及び前記メモリD402を制御する制御プロセッサK403と、前記演算処理装置400と前記メモリD402と前記制御プロセッサK403を接続する高速バスR401とを有する。

【0104】そして、 3×3 の積和演算に必要な係数を $k(x, y)$ [ただし $x=1, 2, 3, y=1, 2, 3$]とし、予め前記制御プロセッサK403により、P0-1、P12-1、P10-1の乗算器・シフターの定数格納用メモリCT1に、前記係数 $k(1, 1)$ 、 $k(1, 2)$ 、 $k(1, 3)$ を格納し、P2-1、P13-1、P14-1の乗算器・シフターの定数格納用メモリCT1に、前記係数 $k(2, 1)$ 、 $k(2, 2)$ 、 $k(2, 3)$ を格納し、P9-1、P6-1、P15-1の乗算器・シフターの定数格納用メモリCT1に、前記係数 $k(3, 1)$ 、 $k(3, 2)$ 、 $k(3, 3)$ を格納しておく。

【0105】次に、前記制御プロセッサK403から 3×3 の積和演算の指示を示すコマンドCMAが与えられたとき、前記メモリD402から読み出すデータを行列の形d(a, b)で表現すると、該演算処理装置400

17

は前記高速バスとの接続をデータd(u, v)がP0に、データd(u, v+2)がP12に、データd(u, v+1)とd(u, v+4)がP3に、データ(u, v+3)がP15に入力するように切り替えられ、さらに該基本処理装置内の出力信号のセクタが、P0の演算結果がP2に、P2の演算結果がP6に、P12の演算結果がP13に、P13の演算結果がP9に、P15の演算結果がP14に、P14の演算結果がP10に、データd(u, v+1)がP3からP2に、P2からP10に、データd(u, v+2)がP12からP14に、P14からP6に、データd(u, v+3)がP15からP13に、データd(u, v+4)がP3からP1に、P1からP9に伝わるように切り替えられる。

【0106】次に、前記メモリD402から前記高速バスR401を経て、データを読みだし、

P0で $S1' = k(1, 1) \times d(u, v)$, $S2' = k(1, 2) \times d(u, v+1)$, $S3' = k(1, 3) \times d(u, v+2)$

を、

P12で $T1' = k(1, 1) \times d(u+2, v)$, $T2' = k(1, 2) \times d(u+2, v+1)$, $T3' = k(1, 3) \times d(u+2, v+2)$

を、

P15で $W1' = k(3, 1) \times d(u+3, v)$, $W2' = k(3, 2) \times d(u+3, v+1)$, $W3' = k(3, 3) \times d(u+3, v+2)$

を、

P2で $S1'' = k(2, 1) \times d(u+1, v) + S1'$, $S2'' = k(2, 2) \times d(u+1, v+1) + S2'$, $S3'' = k(2, 3) \times d(u+1, v+2) + S3'$

を、

P14で $T1'' = k(2, 1) \times d(u+2, v) + T1'$, $T2'' = k(2, 2) \times d(u+2, v+1) + T2'$, $T3'' = k(2, 3) \times d(u+2, v+2) + T3'$

を、

P13で $W1'' = k(2, 1) \times d(u+3, v) + W1'$, $W2'' = k(2, 2) \times d(u+3, v+1) + W2'$, $W3'' = k(2, 3) \times d(u+3, v+2) + W3'$

を、

P6で $S1''' = k(3, 1) \times d(u+2, v) + S1''$, $S2''' = k(3, 2) \times d(u+2, v+1) + S2''$, $S3''' = k(3, 3) \times d(u+2, v+2) + S3''$

及び $S1 = S1''' + S2''' + S3'''$

を、

P10で $T1''' = k(1, 1) \times d(u+1, v)$

18

$+ T1''$, $T2''' = k(1, 2) \times d(u+1, v+1) + T2''$, $T3''' = k(1, 3) \times d(u+1, v+2) + T3''$

及び $T1 = T1''' + T2''' + T3'''$

を、

P9で $W1''' = k(3, 1) \times d(u+4, v) + W1''$, $W2''' = k(3, 2) \times d(u+4, v+1) + W2''$, $W3''' = k(3, 3) \times d(u+4, v+2) + W3''$

及び $W1 = W1''' + W2''' + W3'''$

を計算することにより、9サイクル毎に3種類の結果を得る。

【0107】第2に、演算処理装置400の後段にさらに2段演算処理装置を付加し、前記演算処理装置400内の基本処理装置間の接続として、新たにデータd(u, v)がP0からP1に、P1からP5に、データd(u, v+3)がP13からP5に伝わるようにする。

また、該演算処理装置400間の接続として、前段の演算処理装置400のP6、P9、P10の演算結果を1サイクル遅らせてそれぞれ次段のP0、P12、P15の入力とし、データd(u, v)が前段の演算処理装置400のP5から次段の演算処理装置400のP0へ、同様にデータd(u, v+1)がP10からP3へ、データd(u, v+2)がP6からP12へ、データd(u, v+3)がP5からP15へ、データd(u, v+4)がP9からP3へ伝わるようにする。

さらに、積和演算の係数k(x, y)を、1段目の演算処理装置400については、P0-1、P12-1、P10-1のCT1にk(1, 1)を、P2-1、P13-1、P14-1のCT1にk(1, 2)を、P15-1、P9-1、P6-1のCT1にはk(1, 3)を格納し、2段目の演算処理装置400には、P0-1、P12-1、P10-1のCT1にk(2, 1)を、P2-1、P13-1、P14-1のCT1にk(2, 2)を、P15-1、P9-1、P6-1のCT1にはk(2, 3)を格納し、3段目の演算処理装置400については、P0-1、P12-1、P10-1のCT1にk(3, 1)を、P2-1、P13-1、P14-1のCT1にk(3, 2)を、P15-1、P9-1、P6-1のCT1にはk(3, 3)を格納して演算を行うことにより、1サイクル毎に1種類の結果を得る。

【0108】次に、図14を参照して第8実施例をさらに説明する。

【0109】メモリD402には既に演算に使用するデータが格納されている。データは2次元の配置を持つものであり、このデータの一部を図15に図示する。図16は積和演算の係数を示した3×3のマトリクスである。例えば図15におけるデータu33に対応した積和演算は結果をZで表すと、

【0108】次に、図14を参照して第8実施例をさらに説明する。

【0109】メモリD402には既に演算に使用するデータが格納されている。データは2次元の配置を持つものであり、このデータの一部を図15に図示する。図16は積和演算の係数を示した3×3のマトリクスである。例えば図15におけるデータu33に対応した積和演算は結果をZで表すと、

$Z = k1 \times u22 + k2 \times u23 + k3 \times u24$

19

+k4×u32+k5×u33+k6×u34
+k7×u42+k8×u43+k9×u44

である。上記係数は上記した表現と、

k1=k(1, 1) k2=k(2, 1) k3=k(3, 1)
k4=k(1, 2) k5=k(2, 2) k6=k(3, 2)
k7=k(1, 3) k8=k(2, 3) k9=k(3, 3)

のように対応する。

【0110】制御プロセッサ403は処理するデータをメモリ402に、上記係数を演算処理装置400に設定した後、積和演算を実行するためのコマンド“CMA”に対応するコードを設定し、演算処理装置400に演算スタートの指示を行い、演算を実行させる。

【0111】コマンド“CMA”実行時のデータの流れを図17に示す。

【0112】P0-1、P12-1、P10-1のCT1には“k1”、“k4”、“k7”が、P2-1、P13-1、P14-1のCT1には“k2”、“k5”、“k8”が、P9-1、P6-1、P15-1のCT1には“k3”、“k6”、“k9”が格納されている。これら係数はCT1のアドレス1~3に格納することとしRF60の様に、CT1(1)~CT1(3)と表す事にする。格納された状態を図18に図示する。

【0113】図17の様に、入力データをd(u, v)=a, d(u+1, v)=b, d(u+2, v)=c, d(u+3, v)=d, d(u+4, v)=eで表す。

【0114】1段目の基本処理装置については要素処理装置0は未使用であり、要素処理装置1において入力データとCT1(1)との積が計算され、入力データと演算結果は同一タイミングで2列目に送られる。この演算結果は、

$S1' = k1 \times a$, $T1' = k1 \times c$, $W1' = k3 \times d$, である。

【0115】2段目の基本処理装置も要素処理装置0は未使用である。要素処理装置1で1列目の基本処理装置からの入力データとCT1(1)との積を計算し、 $S1'$ 、 $T1'$ 、 $W1'$ を入力した要素処理装置は、ALU20で積の結果との加算を行い、 $S1''$ 、 $W1''$ 、 $W1''$ とし、3段目の基本処理装置へ出力する。入力データa~eも同じタイミングで出力される。この演算結果は

$S1'' = k2 \times b + S1'$, $T1'' = k2 \times d + T1'$, $W1'' = k2 \times c + W1'$, である。

【0116】3段目の基本処理装置については要素処理装置1は2段目の要素処理装置1と同様の処理を行い、下記演算結果は図6の信号125により要素処理装置0に出力され、ALU20を素通りしてRF60に格納される。演算結果は、

20

$S1''' = k3 \times c + S1''$, $T1''' = k3 \times e + T1''$, $W1''' = k1 \times d + W1''$, である。

【0117】以上が、1組のデータa~eに関しての処理である。この処理はvを1インクリメントしたデータの組が連続的に入力され実行される。乗算に使用する係数はCT1(1)→CT1(2)→CT1(3)→CT1(1)→…のようにローテーションされて使用される。

【0118】従って、 $S1'''$ 、 $T1'''$ 、 $W1'''$ が計算され、要素処理装置0のRF60に格納されたとき、要素処理装置1のレジスタ62には次の組の入力データの計算結果である $S2'''$ 、 $T2'''$ 、 $W2'''$ が格納されている。これらは共に要素処理装置0のALU20のリソースとなり、次のサイクルでALU20により加算が行われ、結果はRF60の同じアドレスに格納される。

【0119】要素処理装置0のALU20で上記加算結果がRF60に格納されたとき、要素処理装置1のレジスタ62には3つ目の組のデータの計算結果である、 $S3'''$ 、 $T3'''$ 、 $W3'''$ が格納されている。これらは次のサイクルで要素処理装置0のALU20でRF60の出力と加算され、最終的な3×3の積和演算結果となって3段目の基本処理装置から出力される。

【0120】3段目の基本処理装置の要素処理装置0のALU20は次のサイクルでは要素処理装置1からの入力を素通りさせ、RF60に格納することで、それまでの演算結果とのつながりを切る事ができる。

【0121】以上の処理を繰り返すことで3サイクルに3つの積和演算結果を求めることができる。

【0122】次にこの演算処理装置を縦に3つ接続し、基本処理装置の接続としては9段となるようにした場合を考える。

【0123】この場合でも以下の点を除いて、動作は同じである。

【0124】異なる点は、1段目の演算処理装置は乗数として前例のCT1(1)の値のみを、2段目の演算処理装置は前例のCT1(2)の値のみを、3段目の演算処理装置はCT1(3)の値のみを使用し、2段目、3段目の演算処理装置へのデータの入力位置は図17の()部も含めた形となるようにし、3段目の基本処理装置の要素処理装置0は使用せず、要素処理装置1の演算結果をそのまま出力し、次段の演算処理装置の入力とし、次段の演算処理装置は前段の演算処理装置の演算結果を一旦RF60に取り込み、次のサイクルで上段の演算処理装置から出力されるデータの乗算結果と加算をし、2段目の基本処理装置へ伝える。

【0125】こうすることで、1サイクルに3つの積和演算を行うことができる。

【0126】以下に、図14を参照して本発明の第9実施例を説明する。

【0127】説明中、2進データはMSBをビット0と呼び、以下LSBにいくに従い、ビット1、ビット2、…と呼ぶことにする。

【0128】本実施例の構成は、データを格納するメモリD402と、該演算処理装置400及び前記メモリD402を制御する制御プロセッサK403と、前記演算処理装置400と前記メモリD402と前記制御プロセッサK403とを接続する高速バスR401を有する。該演算処理装置400内の基本処理装置及び要素処理装置の呼び名を、前記したと同様に呼ぶ事にし、前記メモリD402に256階調の濃淡画像データが格納されているものとする。

【0129】本実施例は、本処理は濃淡画像データの濃度値のヒストグラムを作成するため、同一濃度の画素数をカウントするものである。制御プロセッサK403よりヒストグラム値生成のコマンドが演算処理装置400に与えられると、演算処理装置400内のMCU280により必要な画像データがメモリD402から読み出される処理が開始される。

【0130】1段目の基本処理装置で生成するデータの形式を図19に示す。濃淡値のデータは1つの基本処理装置に対し、2個与える。従って、生成されるデータの画素数は最大2であり、生成するデータのビット14～15、30～31で表す画素数は01Zのとき1個、10Zのとき2個を示す。

【0131】データは $n=1, 2, \dots, 31$ としたとき、要素処理装置0は濃度値が $8n, 1+8n, 2+8n, 3+8n$ に該当するもので1つのデータを生成する。ALU20により入力データの比較を行い、ALU30でデータを生成する。このデータはP1またはP13に出力される。

【0132】濃度値 $4+8n, 5+8n, 6+8n, 7+8n$ に該当するものは要素処理1のALU20の比較結果によりALU30でデータが生成され、P2、P14に対しての出力データとなる。

【0133】2段目の基本処理装置は1段目からのデータを受け取り、3段目の基本処理装置に出力するデータを生成する。このとき生成するデータの形式を図20に示す。データの濃度値は上位ビットが省略されるが、どの基本処理装置で処理を行ったかで、濃度値が分かるため差し支えはない。

【0134】基本処理装置P1、P13は要素処理装置0のALU20でデータのチェックを行い、ALU30で濃度値 $8n, 1+8n$ に関するデータを生成し、要素処理装置1のALU20で同じくデータのチェックを行い、ALU30で濃度値 $2+8n, 3+8n$ に関するデータの作成を行う。

【0135】要素処理装置0で生成したデータはP5へ、要素処理装置1で生成されたデータはP9に出力される。

【0136】基本処理装置P2、P14は要素処理装置0のALU20でデータのチェックを行い、ALU30で濃度値 $4+8n, 5+8n$ に関するデータを生成し、要素処理装置1のALU20で同じくデータのチェックを行い、ALU30で濃度値 $6+8n, 7+8n$ に関するデータの作成を行う。

【0137】要素処理装置0で生成したデータはP6へ、要素処理装置1で生成されたデータはP10に出力される。

【0138】3段目の基本処理装置は次のような動作となる。3段目の基本処理装置が生成するデータの形式を図21に示す。

【0139】P5は要素処理装置0のALU20で入力データのチェックを行い、ALU30で濃度値 $8n$ に対してのデータの作成を行う。2段目からの入力データは3段目の基本処理装置のRF60の2つのエリアを1サイクル毎に交互に使用することで、1サイクル前のデータが消えないようにする。ALU20でのチェックの結果、濃度値 $8n$ に相当する画素が複数あり、作成するデータが2ワードとなる場合は信号112により基本処理装置内のSU150に通知、SU150はこれを演算処理装置400内のMCU280に通知し、1列目、2列目の基本処理装置の動作を停止させる。ALU30のデータはLM0とそのコントローラLMCに渡される。

【0140】要素処理装置1も同様の処理を濃度値 $1+8n$ に対して行う。

【0141】残りのP9、P6、P10も同じであり、P9…0は濃度値 $2+8n$ に対して、P9…1は $3+8n$ に対して、P6…0は $4+8n$ に対して、P6…1は $5+8n$ に対して、P10…0は $6+8n$ に対して、P10…1は $7+8n$ に対しての処理を行う。3列目の基本処理装置からデータを受け取ったLM0～7はLMCにより制御され、濃度値に該当するアドレスの内容に画素のカウント数が加算され更新される。

【0142】以上の動作に必要な各要素処理装置のALU20、30の制御信号はAOP(0:3)の“FNC”とAOP(3:3)の定義番号で指示され、予めこの制御信号に対応してALU20、30が動作するように組み込まれている。

【0143】上記処理を画像データ分行うことで、必要な濃度ヒストグラム値を求める事ができる。一度に読みだしたデータがすべて同じ分類の異なるものである場合は処理速度は最悪となるが、実際には近傍データを読み込むことが多いため、同一濃度の画素が多くなる事が期待できることで、処理速度もそれなりのものが期待できる。

【0144】以下に、図22を参照して本発明の第10実施例を説明する。

【0145】この実施例においては、画像データと定数及び演算中間結果を格納するメモリD402と、演算処

理装置400と、演算処理装置400を制御する制御プロセッサK403と、メモリD402、演算処理装置400、制御プロセッサK403とを接続する高速バスR401とを具備する。そして、予め前記制御プロセッサK403、演算処理装置400により、演算を行う画素のサイズSと、演算に要する画素データの左肩に位置する画素のアドレスAと、アフィン変換に必要な補間の為の各画素値に対応した係数Mを計算し、演算結果を格納するメモリD402のアドレスに合わせて、係数Mの表（以下テーブルと呼ぶ）を作成しておくことにより、演算を行う際に、結果を格納する側のアドレスに従い、アドレスAからサイズSのデータを読みだし、テーブルTの対応する係数Mとの積和演算を行うことによってアフィン変換を行う。

【0146】画像は画素の集合体で表示されるため、アフィン変換を行う場合、結果として生成される画像のものと画素の位置に画素データが存在しない場合があり、この場合は元の画像のデータの近傍の画素データで補正を行うことで対応している。この補正は画素一つ一つに一意的に決定されるものである。サイズSはこの補正に必要な画素の範囲を示すものであり、例えばS=5の場合、5×5の25個の近傍画素が必要であることを示す。

【0147】本例のメモリD402は2つのモジュール0、1に分かれており、それぞれ独立にアクセス可能であることにする。制御プロセッサK403で予め、各画素に対して、サイズSに相当する画素分の補正の為の係数を計算し、処理する画素の順番に対応して、テーブルTを作成し、メモリD402のモジュール1に格納しておく。変換を行うもとの画像データはメモリモジュール0に格納しておく。処理に必要な元の画素データのアドレスは処理する画素の順番に従い、演算処理装置400内に保持される。このときのメモリモジュール、演算処理装置400に保持したデータを図23に示す。410はカメラ411のアナログ信号をデジタル信号に変換しメモリD402に直接書き込みをしたり、メモリD402のデジタル信号をアナログ信号に変換し、モニター402に出力するためのA/D・D/A変換回路である。

【0148】図はサイズS=6とした場合のものである。元の画素データをG(m, n)、変換後の画像データをA(m, n)とし、A(0, 0)を求める為の処理にG(100, 100)を左肩とする正方形のエリアの処理が必要とする場合のものである。メモリD402のモジュール1のアドレス[1000, ...]にG(100, 100)、G(101, 100) ...のデータがアドレス[2000, ...]にG(100, 101)、G(101, 101)、...の様にデータが格納されているものとする。

【0149】図中(A)がA(0, 0)を求める為の補正の係数であり、(C)が演算に使用する元の画像デー

タの代表アドレスである。(B)はA(0, 1)を求める為の補正の係数、(D)はA(0, 1)を求めるための元の画像データの代表アドレスである。

【0150】制御プロセッサK403により処理のスタートが指示されると演算処理装置400はA(0, 0)、A(0, 1)、...と処理を進めていく。制御処理装置400はまず内部に処理を行う順に代表アドレスを保持した表（以下、テーブルAと呼ぶ）の先頭に格納された値をアドレスとしてメモリモジュール0にアクセスし、処理に必要な画像データを読み出す。データは代表アドレスをddddとすると、

dddd ~ dddd+5,
 dddd+1000 ~ (dddd+1000) +5
 dddd+2000 ~ (dddd+2000) +5
 dddd+3000 ~ (dddd+3000) +5
 dddd+4000 ~ (dddd+4000) +5
 dddd+5000 ~ (dddd+5000) +5

のG(n, m)の画像データを読みだし、同時にメモリモジュール1から該当する補正の係数を読み出す。テーブルAの先頭に格納されているのは1000であるから、アドレス1000~1005、2000~2005、3000~3005、5000~5005、6000~6005の値がモジュール0から、メモリモジュール1から(A)が、k1、k2、...の順に読み出され積和演算を行うことで変換後の画素値を算出する。

【0151】以下に図1を参照して本発明の第1実施例を説明する。

【0152】本実施例においては、要素処理装置1において、定数を格納するメモリCT1(11)、CT2(21)、CT3(31)及びRF60と乗算器・シフター10及びALU20、30の間のセレクトの入力に、CS50(信号65)の出力を付加し、前記要素処理装置1の処理が、同一の処理を繰り返し行い、そのための多数の定数が必要な時に、CS50内に前記乗算器・シフター10、ALU20、30の演算に必要な定数を格納しておき、処理に必要な制御信号はCS50の出力を一度PR51に保持したまま、CS50のアドレスを前記定数を格納したアドレスにすることで、CT1(11)、CT2(21)、CT3(31)の大きさを越えるような定数を必要とする演算を行う。図中のセレクト70のリソースBはCS50の出力である。

【0153】例えば、FFTを実行する場合、実部(xr, yr)と虚部(xi, yi)と回転因子(WR, WI)が必要である。回転因子を予めCS50に格納しておき、CU41の出力である信号43でCS50からFFTに必要な制御信号を出力し、PR51に保持し、これによりセレクト70はCS50の出力を選択するように設定され、この状態で、信号92、93で与えられるデータに同期してCS50から対応する回転因子が出力されるように、CU41は信号43を制御する。FFT

の処理の実行中は乗算器・シフター10、ALU20、30の処理は一定であり、PR51の出力の変更は必要ない為、外部から回転因子を取り込む必要がなく処理を行うことができる。

【0154】以上説明したように、本実施例の構成と、これを使用して装置を作成することにより、画像処理等を行う場合、汎用の処理装置でプログラムにより行うことにより時間がかかりすぎるという問題を、特殊用途向けの処理装置となるような規模にせず、相応の処理能力がある装置が作成できるという効果がある。

【0155】

【発明の効果】以上詳述したように、本発明の演算処理装置によれば、接続関係やデータの流れを適宜変更できるので、装置規模を大きくせずに処理能力が向上する。

【図面の簡単な説明】

【図1】第1実施例に係る要素処理装置のブロック図である。

【図2】第4実施例に係る画像処理装置のブロック図である。

【図3】要素処理装置の外部に制御プロセッサを接続した図である。

【図4】制御信号の値の一例を示す図である。

【図5】制御信号の他の例を示す図である。

【図6】本発明の第2実施例に係る基本処理装置のブロック図である。

【図7】基本処理装置の外部に制御プロセッサを接続した図である。

【図8】本発明の第3、7実施例に係る演算処理装置の構成図である。

【図9】演算処理装置の外部に制御プロセッサを接続した図である。

【図10】本発明の第4実施例を説明するための図である。

【図11】本発明の第5実施例に係る基本処理装置のブロック図である。

【図12】本発明の第6実施例の構成を示す図である。

【図13】第7実施例における各要素処理装置の入出力データを示す図である。

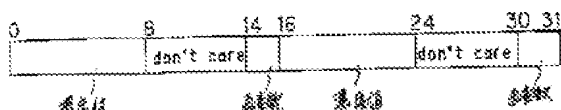
【図14】本発明の第8、9実施例の構成を示す図である。

【図15】メモリDに記憶されたデータの一部を示す図

【図16】

k1	k2	k9
k4	k5	k6
k7	k8	k9

【図19】



である。

【図16】積和演算の係数を示した3×3のマトリクスである。

【図17】コマンド“CMA”実行時のデータの流れを示す図である。

【図18】計数をCT1に格納した状態を示す図である。

【図19】第9実施例におけるデータの形式を示す図である。

10 【図20】第9実施例におけるデータの他の形式を示す図である。

【図21】第9実施例におけるデータの他の形式を示す図である。

【図22】本発明の第10実施例の構成を示す図である。

【図23】第10実施例におけるメモリモジュール及び演算処理装置に保持されたデータを示す図である。

【図24】乗算器・シフターの制御コードを示す図である。

【図25】ALUの制御コードを示す図である。

【図26】セレクトの制御コードを示す図である。

【図27】定数メモリの制御コードを示す図である。

【図28】レジスタRFの制御コードを示す図である。

【図29】データ整備回路の制御コードを示す図である。

【図30】制御信号133の意味を表す図である。

【図31】制御信号151、152の意味を表す図である。

【図32】RFの入出力信号とポートとの関係を示す図である。

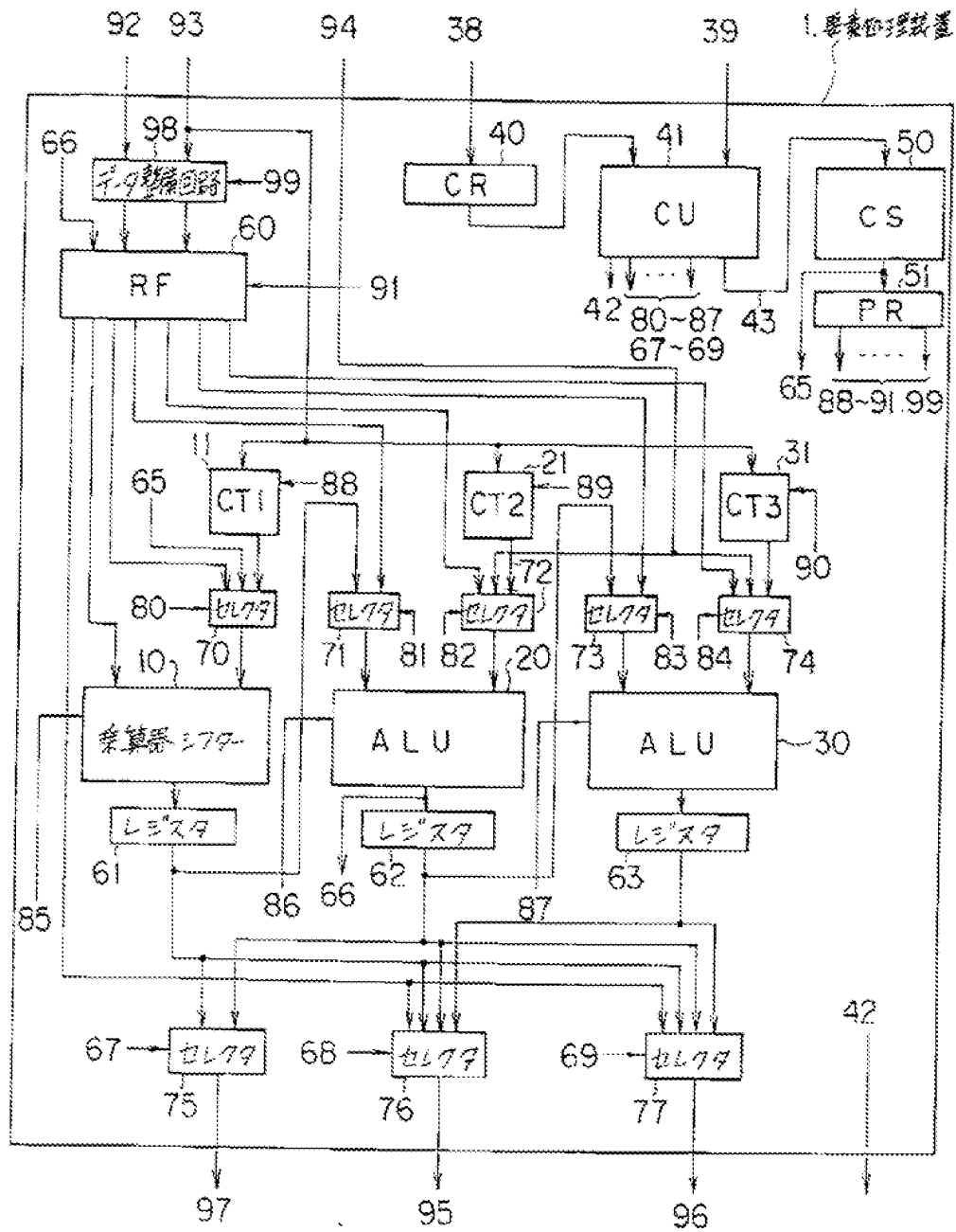
【符号の説明】

1…要素処理装置、10…乗算器・シフター、11…定数格納用メモリCT1、20…ALU20、21…定数格納用メモリCT2、30…ALU30、31…定数格納用メモリCT3、40…レジスタCR、41…制御部CU、50…メモリCS、51…ラッチPR、60…レジスタRF、61…レジスタ、62…レジスタ、63…レジスタ、70…セレクト、71…セレクト、72…セレクト、73…セレクト、74…セレクト、75…セレクト、76…セレクト、77…セレクト。

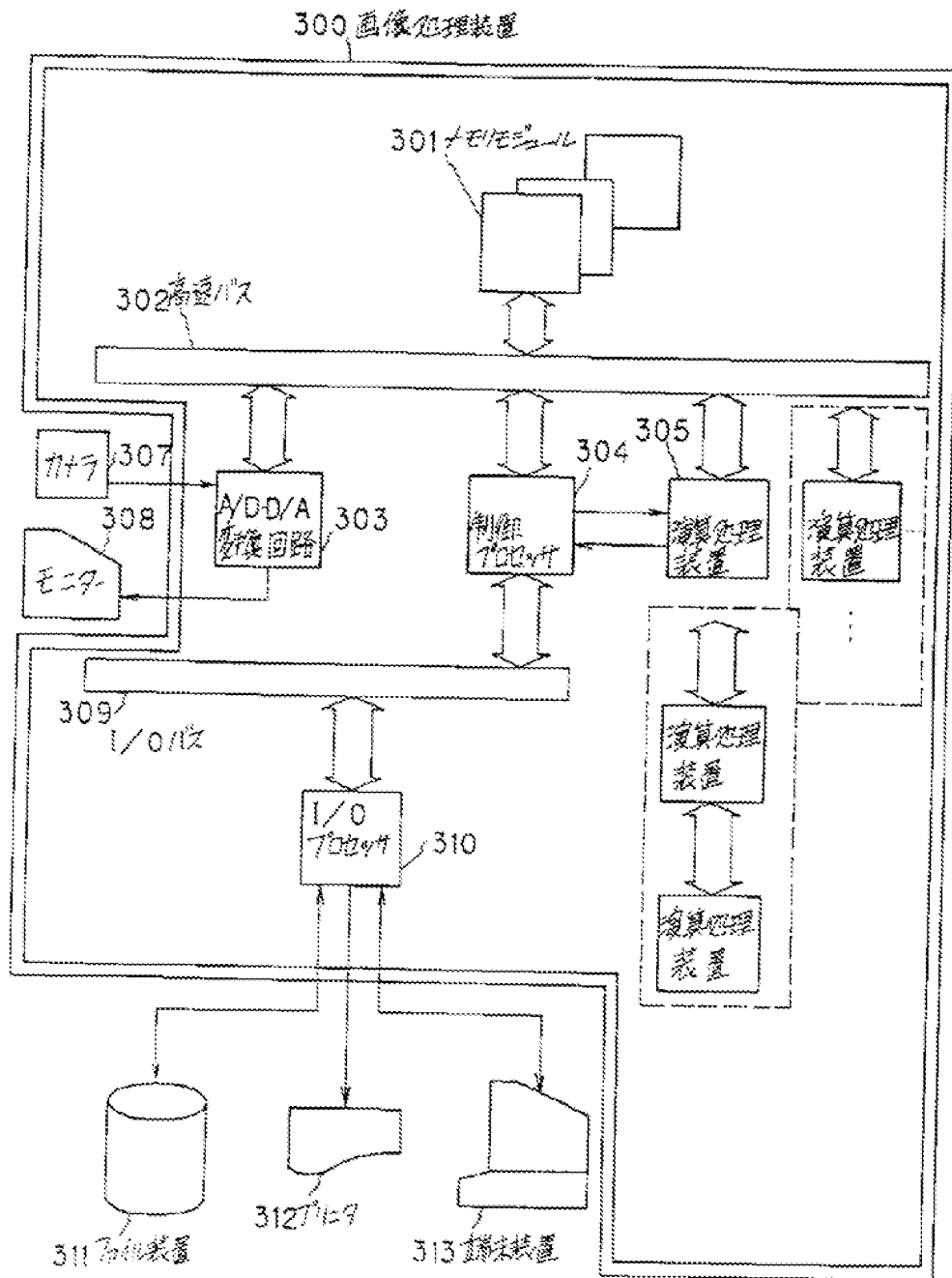
【図29】

FOP(0,2)	意味
00	スルー
01	反転
10	符号おぎ
11	定数表

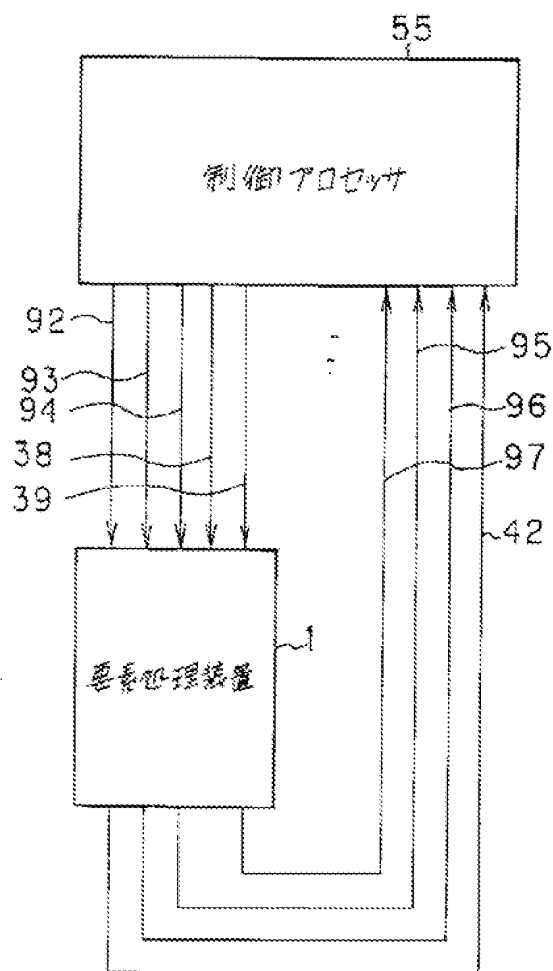
【図1】



【図2】



【図3】



【図4】

00	110000	001000	000000	00	000	000	01	10
①	②	③	④	⑤	⑥	⑦	⑧	⑨

- ① … FOP(0:2), 信号92の値
 ② … NOP(0:6), 信号93の値
 ③ … ACP(0:6), 信号94の値
 ④ … ACP(0:6), 信号87の値 ALU30は空欄に0を出力
 ⑤ … S3OP(0:2), 信号80の値 11/2A空欄に00Z
 ⑥ … S2OP(0:1), 信号81の値 11/2A空欄に0Z
 ⑦ … S3OP(0:2), 信号82の値 11/2A空欄に00Z
 ⑧ … S2OP(0:1), 信号83の値 空欄に0Z
 ⑨ … S3OP(0:2), 信号84の値 空欄に0Z
 ⑩ … S2OP(0:1), 信号85の値 空欄に0Z
 ⑪ … S3OP(0:2), 信号86の値 11/2B空欄に01Z
 ⑫ … S3OP(0:2), 信号87の値 11/2C空欄に01Z

【図15】

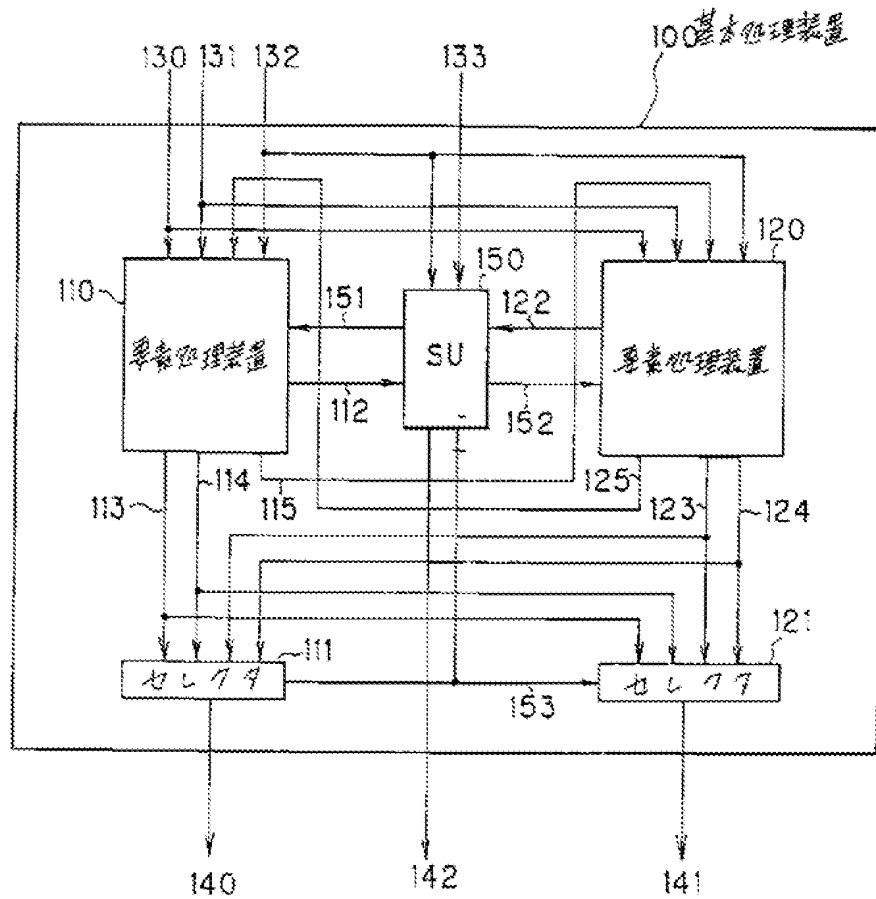
U11	U12	U13	U14	U15
U21	U22	U23	U24	U25
U31	U32	U33	U34	U35
U41	U42	U43	U44	U45
U51	U52	U53	U54	U55

【図5】

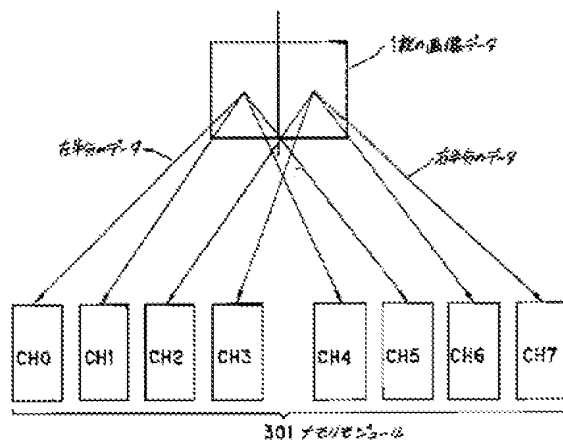
(A)...	0011	001	010	1001	011	1010	000	00000	00000	00000
	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩
(B)...	1011	001	010	1001	011	1010	000	00000	00000	00000

- ① … ROP(0:4), 信号a RF(3)の値、信号27の値
 ② … ROP(1:3), 信号b RF(1)の値
 ③ … ROP(1:3), 信号c RF(2)の値
 ④ … ROP(0:4), 信号d RF(1)の値、信号27の値、信号91
 ⑤ … ROP(1:3), 信号e RF(3)の値
 ⑥ … ROP(0:4), 信号f RF(2)の値、信号27の値
 ⑦ … ROP(1:3), 信号g 空欄に0を出力
 ⑧ … TOP(0:5), CT1空欄に0を出力、信号88
 ⑨ … TOP(0:5), CT2空欄に0を出力、信号89
 ⑩ … TOP(0:5), CT3空欄に0を出力、信号90

【図6】



【図10】



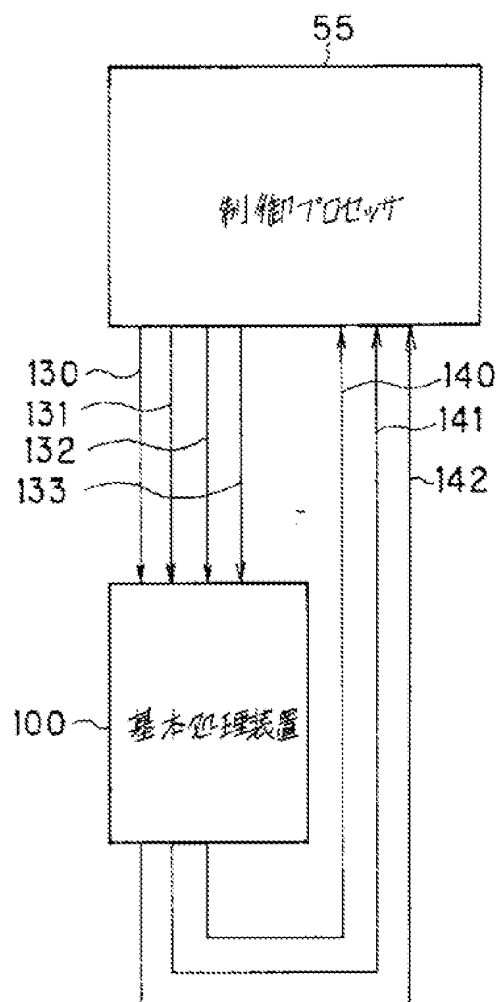
【図26】

AOP(10:3)	意味	AOP(3:2)	意味
000	NOP	00	32ビット演算
001	ADD	01	16ビット演算
010	AND	10	8ビット演算
011	SUB	11	不定長
100	OR		
101	EXOR		
110	ENCR		
111	FNC		

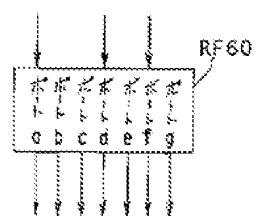
AOP(15)	意味
	AOP#000Z'111Z'
0	オペランドの半分の値を保持
1	オペランドの半分の値を保持

AOP(10:3)はALUに結合した演算処理であり、AOP(3:2)はALUに結合した演算処理のビット幅を示す。

【図7】

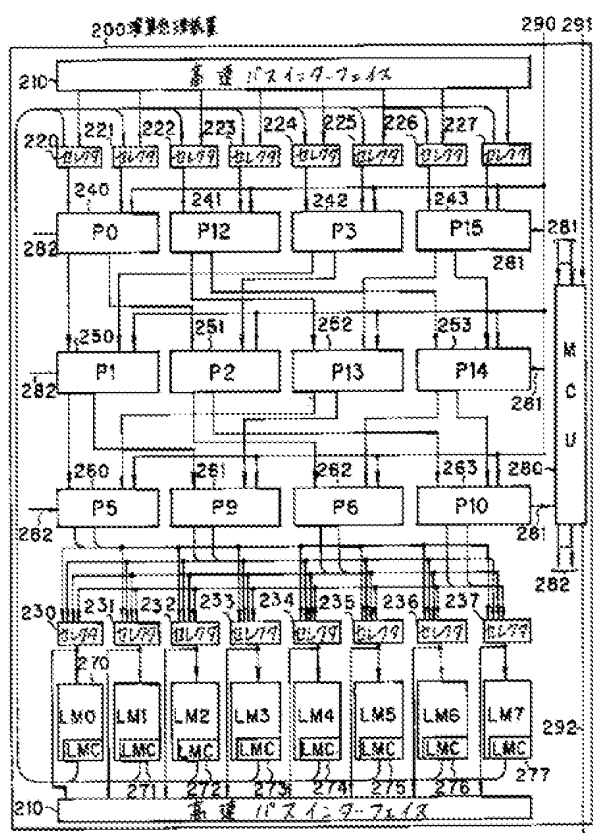


【図32】

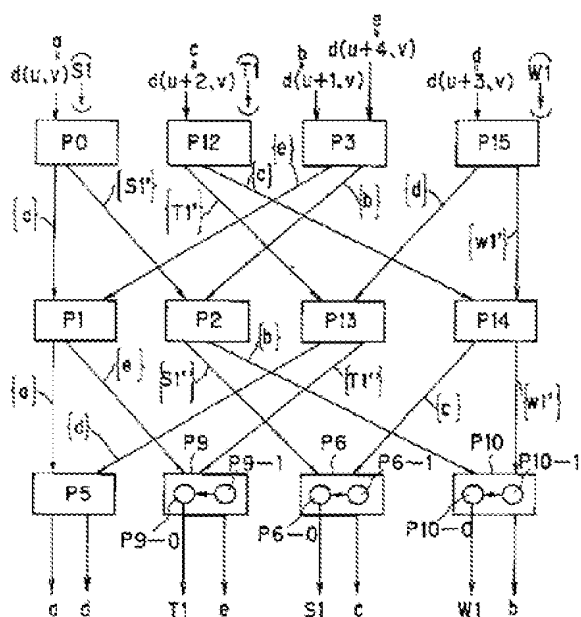


(32は図1の接続と対応)

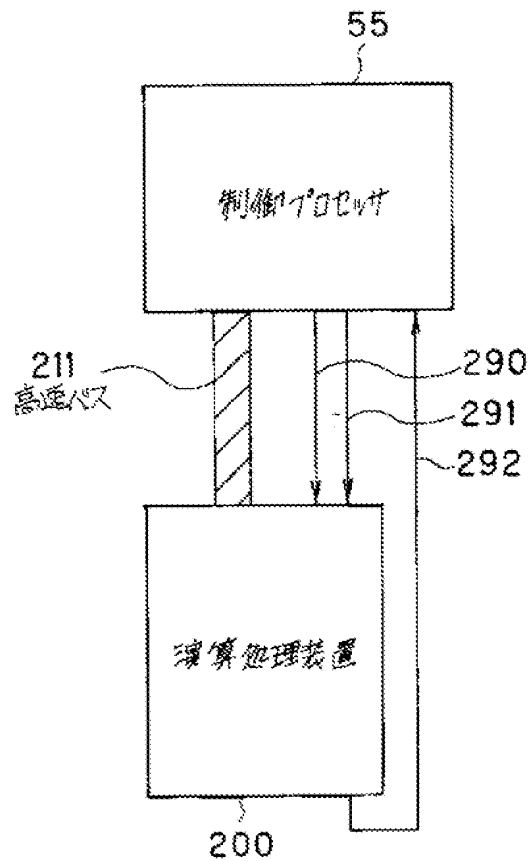
【図8】



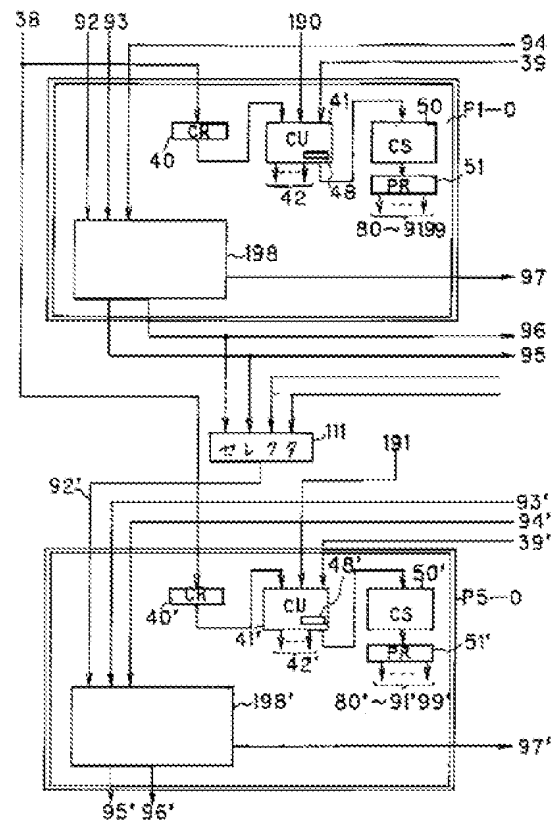
【図17】



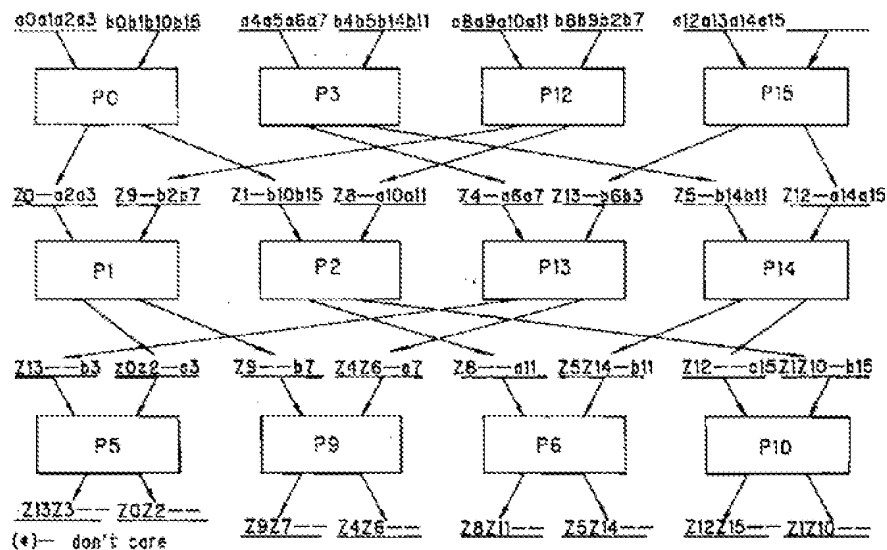
【図9】



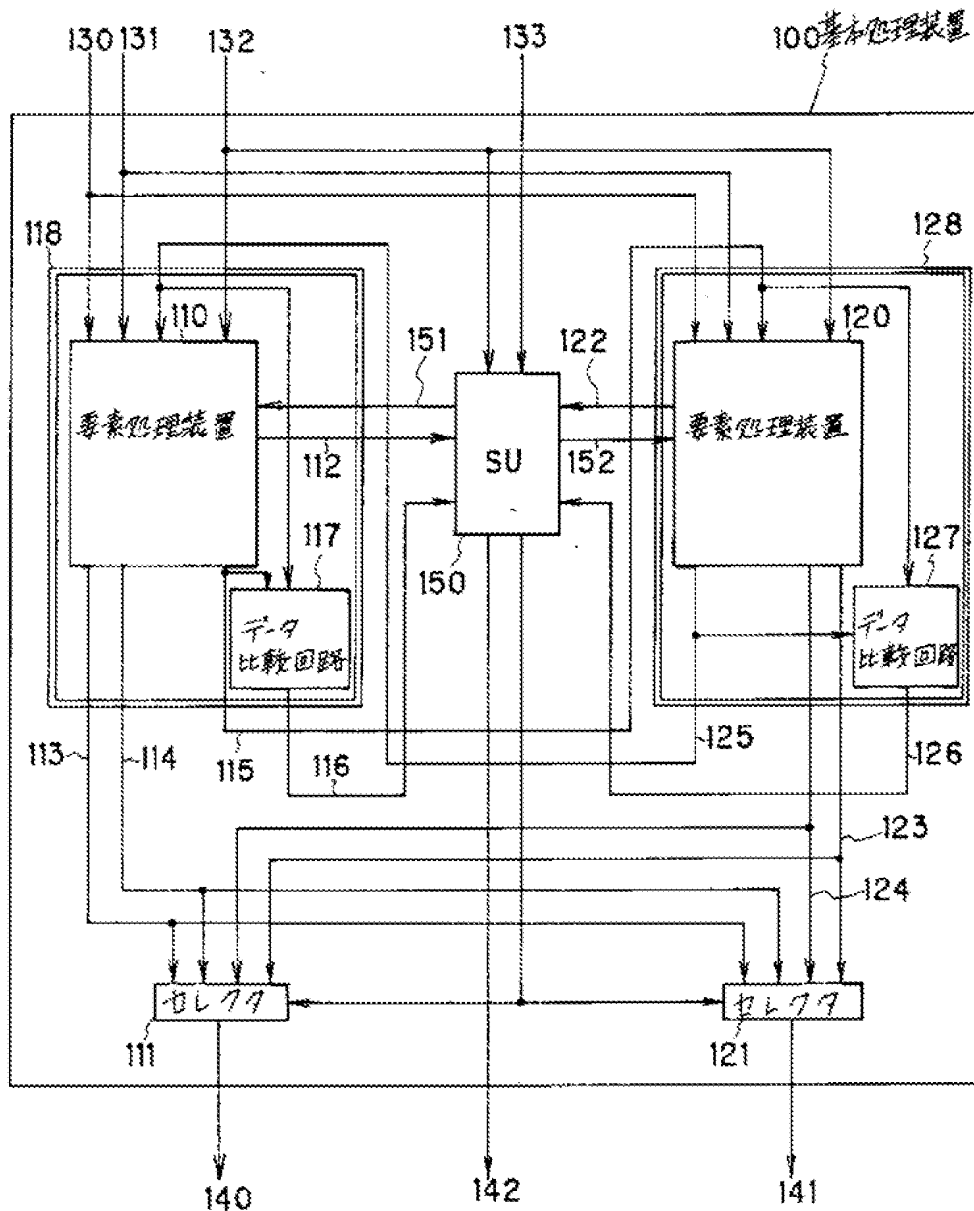
【図12】



【図13】



【図11】



【図26】

【図27】

S20P(0:1)	出力	S30P(0:2)	出力
0	A	00	A
1	B	01	B
		10	C
		11	D(C)

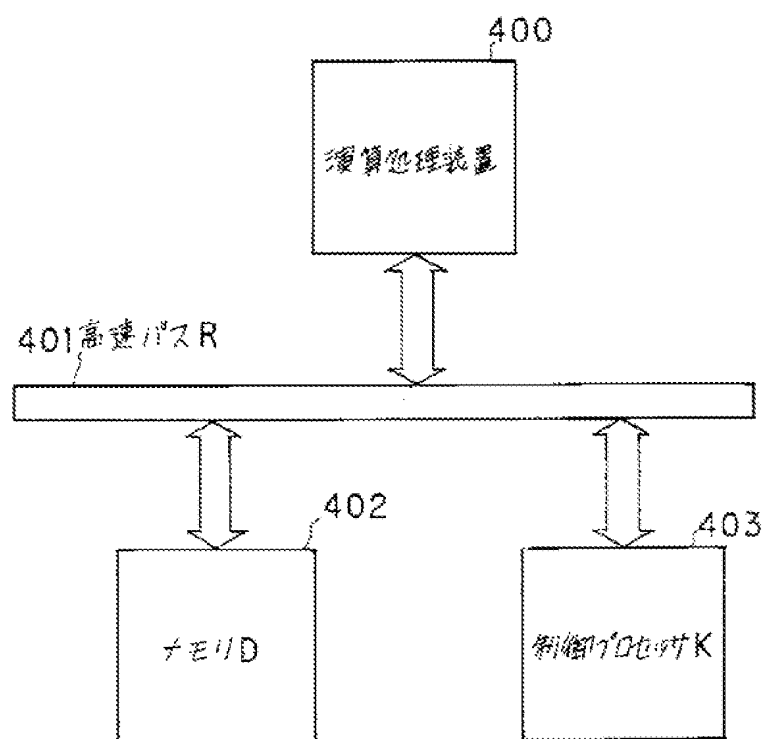
リソース ~ ↑ ↓ ...
セレクト ~ A B ...
出力 ~ ↑
(X) リソース 3 未満の場合

TOP(0:1)	意味	TOP(1:4)	意味
0	読み出し	xxxx	アドレス
1	書き込み		

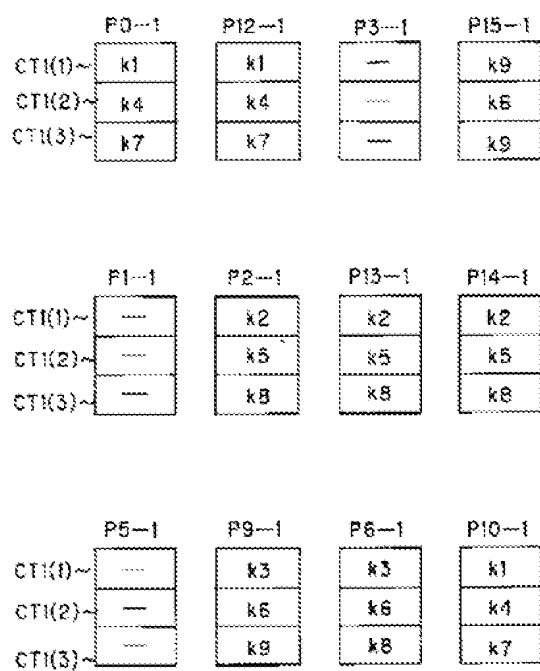
【図28】

RGP(0:1)	意味	RGP(1:3)	意味
0	読み出し	xxxx	アドレス
1	書き込み		

【例 14】



【例 18】

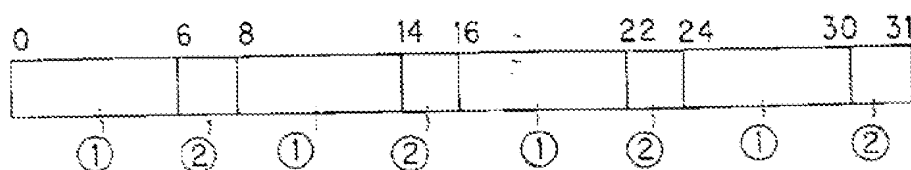


(*) — 付箋使用

【图 2-3】

FPGAの構成		Xilinxの構成		FPGAの構成	
FPGAの構成		Xilinxの構成		FPGAの構成	
0000	0000 (C)	0100	k1 k2 k3 k4	0000	0000 (C)
0001	0001 (D)	0101	k5 k6 k7 k8	0001	0001 (D)
0002	0002 (E)	0102	k9 k10 k11 k12	0002	0002 (E)
0003	0003 (F)	0103	k13 k14 k15 k16	0003	0003 (F)
0004	0004 (G)	0104	k17 k18 k19 k20	0004	0004 (G)
0005	0005 (H)	0105	k21 k22 k23 k24	0005	0005 (H)
0006	0006 (I)	0106	k25 k26 k27 k28	0006	0006 (I)
0007	0007 (J)	0107	k29 k30 k31 k32	0007	0007 (J)
0008	0008 (K)	0108	k33 k34 k35 k36	0008	0008 (K)
0009	0009 (L)	0109	k37 k38 k39 k40	0009	0009 (L)
0010	0010 (M)	0110	k41 k42 k43 k44	0010	0010 (M)
0011	0011 (N)	0111	k45 k46 k47 k48	0011	0011 (N)
0012	0012 (O)	0112	k49 k50 k51 k52	0012	0012 (O)
0013	0013 (P)	0113	k53 k54 k55 k56	0013	0013 (P)
0014	0014 (Q)	0114	k57 k58 k59 k60	0014	0014 (Q)
0015	0015 (R)	0115	k61 k62 k63 k64	0015	0015 (R)
0016	0016 (S)	0116	k65 k66 k67 k68	0016	0016 (S)
0017	0017 (T)	0117	k69 k70 k71 k72	0017	0017 (T)
0018	0018 (U)	0118	k73 k74 k75 k76	0018	0018 (U)
0019	0019 (V)	0119	k77 k78 k79 k80	0019	0019 (V)
0020	0020 (W)	0120	k81 k82 k83 k84	0020	0020 (W)
0021	0021 (X)	0121	k85 k86 k87 k88	0021	0021 (X)
0022	0022 (Y)	0122	k89 k90 k91 k92	0022	0022 (Y)
0023	0023 (Z)	0123	k93 k94 k95 k96	0023	0023 (Z)
0024	0024 (A)	0124	k97 k98 k99 k100	0024	0024 (A)
0025	0025 (B)	0125	k101 k102 k103 k104	0025	0025 (B)
0026	0026 (C)	0126	k105 k106 k107 k108	0026	0026 (C)
0027	0027 (D)	0127	k109 k110 k111 k112	0027	0027 (D)
0028	0028 (E)	0128	k113 k114 k115 k116	0028	0028 (E)
0029	0029 (F)	0129	k117 k118 k119 k120	0029	0029 (F)
0030	0030 (G)	0130	k121 k122 k123 k124	0030	0030 (G)
0031	0031 (H)	0131	k125 k126 k127 k128	0031	0031 (H)
0032	0032 (I)	0132	k129 k130 k131 k132	0032	0032 (I)
0033	0033 (J)	0133	k133 k134 k135 k136	0033	0033 (J)
0034	0034 (K)	0134	k137 k138 k139 k140	0034	0034 (K)
0035	0035 (L)	0135	k141 k142 k143 k144	0035	0035 (L)
0036	0036 (M)	0136	k145 k146 k147 k148	0036	0036 (M)
0037	0037 (N)	0137	k149 k150 k151 k152	0037	0037 (N)
0038	0038 (O)	0138	k153 k154 k155 k156	0038	0038 (O)
0039	0039 (P)	0139	k157 k158 k159 k160	0039	0039 (P)
0040	0040 (Q)	0140	k161 k162 k163 k164	0040	0040 (Q)
0041	0041 (R)	0141	k165 k166 k167 k168	0041	0041 (R)
0042	0042 (S)	0142	k169 k170 k171 k172	0042	0042 (S)
0043	0043 (T)	0143	k173 k174 k175 k176	0043	0043 (T)
0044	0044 (U)	0144	k177 k178 k179 k180	0044	0044 (U)
0045	0045 (V)	0145	k181 k182 k183 k184	0045	0045 (V)
0046	0046 (W)	0146	k185 k186 k187 k188	0046	0046 (W)
0047	0047 (X)	0147	k189 k190 k191 k192	0047	0047 (X)
0048	0048 (Y)	0148	k193 k194 k195 k196	0048	0048 (Y)
0049	0049 (Z)	0149	k197 k198 k199 k200	0049	0049 (Z)

【図20】



① --- 濃度対応値. それぞれの値が、下表の如く濃度値に対応する。
下表におけるmは濃度対応値とする。

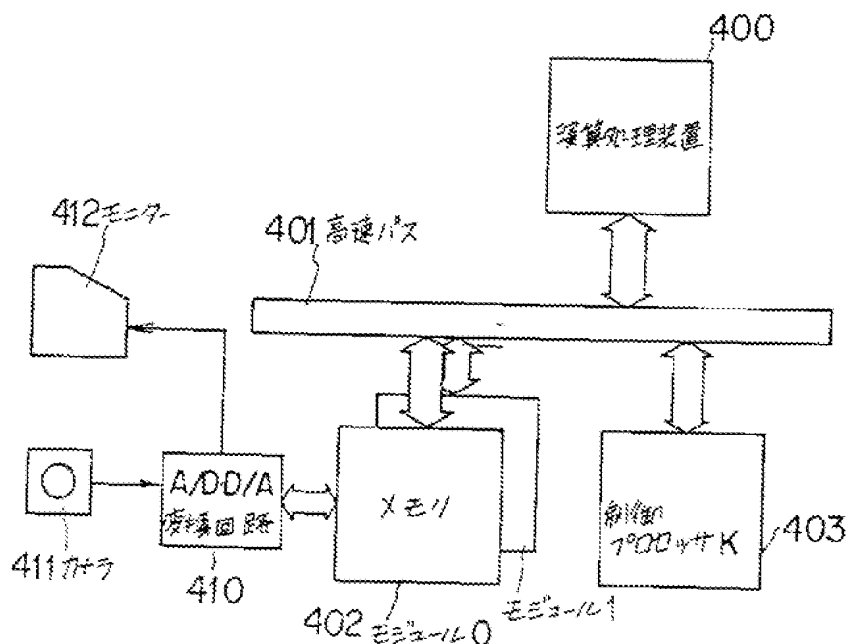
	P5出力側の濃度値	P9出力側の濃度値	P6出力側の濃度値	P10出力側の濃度値
m=0~31	8m	2+8m	4+8m	6+8m
m=32~63	1+8m	3+8m	5+8m	7+8m

② --- 画素数を示す

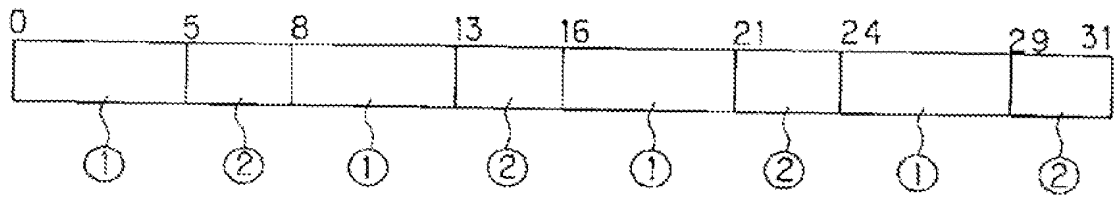
値	00	01	10	11
画素数	1	2	3	4

(注) 濃度対応値が小さい値から読み取れる。従ってポート1,2,3で濃度対応値が"0"となるポート番号が千分点となる。

【図22】



【図21】



①… 乗数付加値

	LM0 付加値	LM1付加	LM2付加	LM3付加	LM4付加	LM5付加	LM6付加	LM7付加
m = 0~31	8m	1+8m	2+8m	3+8m	4+8m	5+8m	6+8m	7+8m

②… 画素数を示す

値	000	001	010	011	100	101	110	111
画素数	1	2	3	4	5	6	7	8

【図24】

MOP(0:2)	意味	MOP(2:2)	意味
0 0	NOP	00	32ビット演算
0 1	左シフト	01	16ビット演算
1 0	右シフト	10	8ビット演算
1 1	乗算	11	未定義

MOP(4:2)	意味	
	MOP(0:2)="11Z"	MOP(0:2)="{01Z or 10}"
0 0	オーバーフロー検出	0-セット
0 1	オーバーフロー未検出	0挿入
1 0	オーバーフロー検出	1挿入
1 1	オーバーフロー未検出	未定義

【図30】

信号値	意 味
000	要素処理装置 110、120 の動作停止
001	要素処理装置 110 の動作停止
010	要素処理装置 120 の動作停止
011	動作開始
100	要素処理装置 110、120 のコマンド設定
101	要素処理装置 110 のコマンド設定
110	要素処理装置 120 のコマンド設定
111	don't care

【図31】

信号値	意 味
00	動作停止
01	コマンド設定
10	動作開始
11	don't care